Exploring near-term quantum applications with graph states from quantum emitters

MSc in Physics, 2021-2022

Love Pettersson

Supervisors:

Peter Lohdal and Stefano Paesani

University of Copenhagen

FACULTY:	Faculty of Science	
INSTITUTE:	The Niels Bohr Institute	
AUTHOR:	Love Pettersson	KU-ID: bdt697
TITLE:	Exploring near-term quantum applications with graph states from quantum emitters	
SUPERVISOR:	Peter Lohdal	peter.lohdal@nbi.ku.dk
CO-SUPERVISOR:	Stefano Paesani	stefano.paesani@nbi.ku.dk
HANDED IN:	May, 2022	

UNIVERSITY OF
COPENHAGEN

Abstract

The heart of this thesis is comprised of a spin-photon interface which can deterministically generate graph states. With this interface we explore the class of graph states available up to eight photons. Further, we construct a realistic error model accounting for its infidelities, which are directly linked to experimental parameters. With the error model, we explore three different quantum algorithms and two error correction applications that can be implement on the available graph states.

More in detail, we analyse Grover's search and Deutsch's algorithm implemented on four qubit graph states. Furthermore, we also describe a variational quantum eigensolver (VQE) which can be implement with the interface, and simulate its performance on a few-qubit experiment. Moreover, we study error-protected measurements of logical qubits, including both Pauli errors and photon loss. This error-protection is investigated in the context of two potentially near term applications: (1) Reading out the spin using graph codes, (2) a BB84 protocol using graph codes instead of single photons.

From our analysis we identify that Grover's and Deutsch's algorithms show significant resilience to errors present in the system, while the VQE protocol appears more susceptible. We then also identify various quantum error-correcting states that can provide noise-suppression, both for losses and Pauli errors, for realistic noises in quantum emitters. These results provide a class of protocols that could enable interesting quantum experiments and applications in near-term spin-photon quantum hardware.

Acknowledgements

First and foremost I would like to thank my supervisor Peter Lohdal for giving me the opportunity to do this project. I truly value that Peter, without hesitation, gave me a theory project even though he is running an experimental group. Also I have very much appreciated the chance of being part of his group.

I would also like to thank my co-supervisor Stefano Paesani. I am more than grateful for the time he took guiding me through this project and his pedagogical feedback. Also his genuine enthusiasm for physics is truly contagious.

Finally I would like to thank all people of Hy-Q for creating such a great atmosphere, and Tom Bell and Alex Jones for fruitful discussions on loss-tolerant encodings.

Contents

Abbreviations

QD: Quantum dot

QECC: Quantum error-correcting code

CNOT: Controlled-not gate

SSM: Star-spin-middle

SSL: Star-spin-leaf

FC: Fully-connected

LC: Local complementation

LC-equivalent: Local Clifford equivalent

PCW: Photonic crystal waveguide

MBQC: Measurement based quantum computing

NISQ: Noisy Intermediate Scale Quantum

QPU: Quantum processor unit

CPU: Central processor unit

VQE: Variational quantum eigensolver

P-VQE: Photonic-variational quantum eigensolver

MUB: Mutually unbiased bases

SKR: Secret key rate

GHZ: Greenberger–Horne–Zeilinger

Chapter 1

Introduction

In this chapter we present some theoretical background to the two main applications considered in this thesis, quantum computing and quantum error correction. In terms of quantum computation, we introduce Deutsch's algorithm, Grover's search algorithm and the variational quantum eigensolver. Further, we will introduce the key aspects of quantum error correction, with special focus on stabilizer codes, which will be important to describe the theory of graph codes.

1.1 The qubit and quantum computation

The *qubit* is the carrier of one unit of quantum information. Unlike its classical counterpart, the *bit*, the qubit can be in a superposition of 0 and 1: $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. This is what allows *quantum parallelism* [1], which potentially offers enormous computational speed up. Manipulating the qubit is realised through unitary operators $\hat{U}$, also referred to as *gates*. A unitary operator fulfils $\hat{U}\hat{U}^\dagger = \hat{U}^\dagger\hat{U} = 1$, where "$\dagger$" indicates the *hermitian adjoint*.

A qubit can be viewed as a normalised vector on a sphere, called the Bloch sphere. This is illustrated in figure 1.1. Any single qubit operation can be defined as a rotation of the normalised vector on the Bloch sphere, and is thus an Euler rotation [3]. Recall, that an Euler rotation is an arbitrary rotation in three dimensions using three consecutive rotations around only two different axes

$$\hat{R}(\alpha, \beta, \gamma) = e^{\frac{-i\hat{Z}\alpha}{2}} e^{\frac{-i\hat{Y}\alpha}{2}} e^{\frac{-i\hat{Z}\alpha}{2}}, \tag{1.1}$$

where $\hat{Y}$ and $\hat{Z}$, along with $\hat{X}$, are the Pauli operators defined as

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \text{and} \quad \hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.2}$$

Hence, by access to rotations around two different axes all single qubit gates are unlocked.

To make a useful *quantum computation*, a register of many qubits is needed. Furthermore, to be able to perform any quantum computation, one needs to be able to access all N-

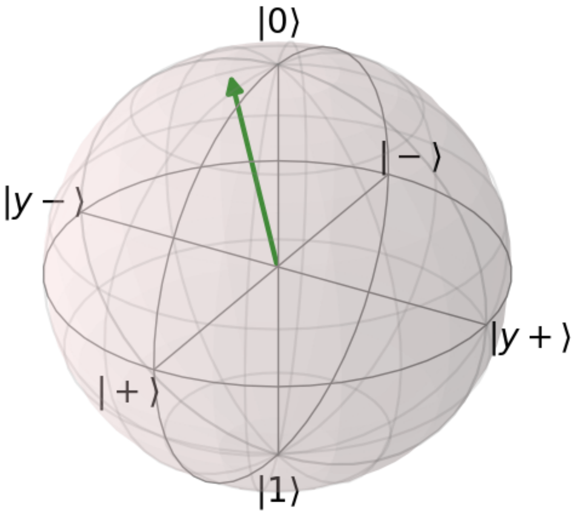
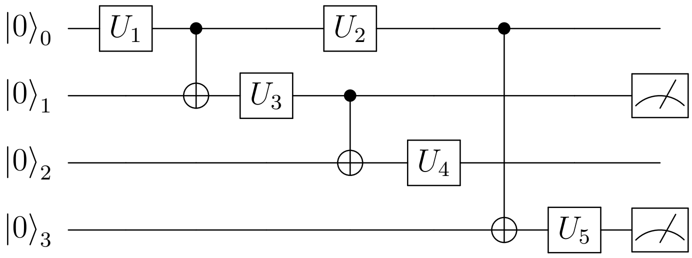


Figure 1.1: The figure illustrates the Bloch sphere with a qubit symbolised by a green arrow. The eigenstates of all axes are shown, with $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$ and $|y\pm\rangle = \frac{|0\rangle \pm i|1\rangle}{\sqrt{2}}$. This was plotted using *Qutip* [2].

qubit unitaries $\hat{U}^{\otimes N}$ on the N-qubit register [4]. This requirement is reduced to requiring access to only one two-qubit gate and all single-qubit gates, which is called a *universal gate set*. This is because all $\hat{U}^{\otimes N}$ can be decomposed by this set of gates [5]. Further, to infer the result of a quantum computation, one needs to perform at least one measurement, which inevitably collapses any superpositions. Constructing a quantum algorithm, one seeks to exploit quantum parallelism while avoiding its collapse from measurement.

In the circuit model of quantum computing, the evolution of the qubits during a quantum computation is readily described by a quantum circuit. A quantum circuit depicts the gates that are needed to perform the computation and they are ordered in time. An example is illustrated below, which consider single qubit gates ranging from one to five, three control gates (two-qubit gates) and two measurements. Here, U_1 is performed first, then the control gate between qubit 0 and 1, etc.



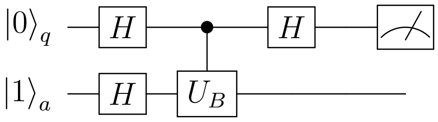
1.2 Quantum algorithms

This section introduces three important quantum algorithms: Deutsch's algorithm [6], Grover's search algorithm [7] and the variational quantum eigensolver [8]. Along with the algorithms themselves, concepts such as an orcale and algorithmic complexity are introduced. We here focus on a high-level theoretical description, while an actual physical implementation of all these algorithms is studied in chapter 4.

1.2.1 Deutsch's algorithm

Deutsch's algorithm aims at solving the following problem: Given a function $F(x) : \{0,1\} \longrightarrow \{0,1\}$, determine if the function is constant or balanced. That is, is $F(0) = F(1)$? Deutsch's algorithm solves this problem by implementing a *black box* that calculates the function $F(x)$. In general, the black box, also called the *oracle*, is designed to solve a certain problem in a single call and is a tool to simplify the calculation of an algorithm's complexity. The complexity of an algorithm is subsequently determined by the number of calls that are needed to the black box to finish the algorithm. Classically two function calls are needed to solve Deutsch's algorithm[1] while, as we shall demonstrate, a quantum computer only needs one.

The circuit below implements Deutsch's algorithm.



where q stands for query qubit and a ancilla qubit. Furthermore, $\hat{H}$ is the Hadamard gate

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{1.3}$$

and $\hat{U}_B$ is our black box, which implements the following controlled-unitary transformation:

$$\hat{U}_B : |i\rangle_q |j\rangle_a \longrightarrow |i\rangle_q |j \oplus F(i)\rangle_a . \tag{1.4}$$

The evolution of the circuit up until the measurement follows:

$$|0\rangle_q |0\rangle_a \xrightarrow{\hat{H}_q \otimes \hat{H}_a} \frac{(|0\rangle_q + |1\rangle_q)(|0\rangle_a - |1\rangle_a)}{2} \xrightarrow{\hat{U}_B} \tag{1.5}$$

$$\frac{(((-1)^{F(0)} + (-1)^{F(1)}) |0\rangle_q + ((-1)^{F(0)} - (-1)^{F(1)})) |1\rangle_r (|0\rangle_a - |1\rangle_a)}{2}. \tag{1.6}$$

[1] Since the function must be evaluated for both inputs to determine if it is constant or balanced.

Hence, the measurement yields $|0\rangle$ ($|1\rangle$) with unit probability for a constant (balanced) function, showing in a very direct way a quantum advantage for this very simple task.

1.2.2 Grover's search algorithm

Grover's search algorithm is a quantum algorithm which offers a quadratic speedup for an unstructured database search [9]. Given an unstructured database with $N = 2^n$ items, Grover's algorithm aims to find one item indexed b_0. Classically, the solution is found through an exhaustive search, which on average finds the solution after $\mathcal{O}(\frac{N}{2})$ steps [7]. The black box $\hat{U}_B$ in Grover's algorithm is composed of two operations

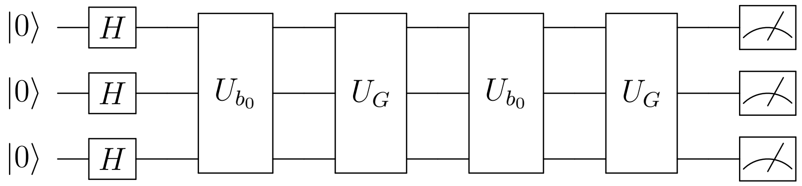
$$\hat{U}_B = \hat{U}_G \hat{U}_{b_0}, \tag{1.7}$$

with $\hat{U}_G$ and $\hat{U}_{b_0}$ defined as

$$\hat{U}_G = 2\,|\psi\rangle\,\langle\psi| - \hat{\mathbb{1}}, \tag{1.8}$$

$$\hat{U}_{b_0} = \hat{\mathbb{1}} - 2\,|b_0\rangle\,\langle b_0|\,, \tag{1.9}$$

where $\hat{\mathbb{1}}$ is the identity matrix and $|b_0\rangle$ is the item to be tagged. For $N = 8$ items, three qubits are needed to compose the database ($2^3 = 8$) and the quantum circuit for implementing the algorithm is illustrated below.



The algorithm is performed in three steps. First the qubit register is initialised in an equal superposition of $N = 2^n$ bit numbers b [9]

$$|\psi\rangle = \hat{H}^{\otimes n}\,|0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{b=0}^{N-1} |b\rangle\,. \tag{1.10}$$

Next, the first operation of the black box, $\hat{U}_{b_0}$ is applied. This operator can be seen as flipping the phase of the item to be tagged b_0:

$$\hat{U}_{b_0}\,|b \neq b_0\rangle = |b\rangle\,, \tag{1.11}$$

$$\hat{U}_{b_0}\,|b_0\rangle = -\,|b_0\rangle\,. \tag{1.12}$$

In the final step we apply the second operation of the black box U_G, which transforms the state in (1.10) as

$$\hat{U}_G\,|\psi\rangle = \sum_{b=0}^{N-1} c'_b\,|b\rangle\,, \tag{1.13}$$

where the transformed coefficients c'_b are given by

$$c'_b = \frac{2}{N} \sum_{j=0}^{N-1} c_j - c_b. \tag{1.14}$$

Denoting

$$\bar{c} = \frac{1}{N}\sum_{b=0}^{N-1} c_b, \tag{1.15}$$

as the average of all the coefficients, the $\hat{U}_G$ operation can be viewed as performing an *inversion around the mean*

$$c_b \longrightarrow c_b' = \bar{c} - (c_b - \bar{c}). \tag{1.16}$$

This increases, after the first iteration, the tagged item's probability amplitude to

$$c_{b_0} \approx \frac{3}{\sqrt{N}}. \tag{1.17}$$

Hence, by repeatedly applying the black-box operation the probability of measuring the tagged item increases. After iterating $\mathcal{O}(\sqrt{N})$ times $b_0 \approx 1$ and measuring all register qubits will give the correct bit number b_0 with next to unit probability.

1.2.3 Variational quantum eigensolver

A promising application on Noisy Intermediate Scale Quantum (NISQ) quantum systems [10] is the variational quantum eigensolver (VQE) [11]. This is because VQE could in principle offer a polynomial time solution to a problem whose classical counter part scales exponentially, all the while having shown to be resilient to noise [11, 12]. The latter being important for NISQ systems and the former being important because it would give the VQE a substantial advantage over classical systems.

The VQE, first developed by Ref. [8], aims to find the ground state energy E_0 of a given Hamiltonian $\hat{H}$. It achieves this by using the variational principle, which given a trial wavefunction $|\psi\rangle$ gives an upper bound to the ground state energy of $\hat{H}$ [3]

$$\frac{\langle\psi|\,\hat{H}\,|\psi\rangle}{\langle\psi|\psi\rangle} \geq E_0. \tag{1.18}$$

Scanning different trial wavefunctions, it is in principle possible to find the true ground state energy. The VQE aims to find a parametrisation of the wavefunction anstatz $|\psi(\phi_1, .., \phi_N)\rangle$, such that (1.18) can be optimised with respect to them. The VQE subsequently performs the optimisation in a hybrid approach between a quantum processor unit (QPU) and a classical central processor unit (CPU), which is illustrated in figure 1.2. The QPU calculates the expectation value of the Hamiltonian while the CPU performs the optimisation step,

$$E(\boldsymbol{\phi}) = \mathbf{min}_{\boldsymbol{\phi}} \frac{\langle\psi(\phi_1, .., \phi_N)|\,\hat{H}\,|\psi(\phi_1, .., \phi_N)\rangle}{\langle\psi(\phi_1, .., \phi_N)|\psi(\phi_1, .., \phi_N)\rangle}, \tag{1.19}$$

feeding a new set of $\boldsymbol{\phi} = \{\phi_i\}$ to the QPU. The power of the VQE lies in the calculation of (1.18) using the quantum processor compared to calculating (1.18) classically [11].

The ansatz, in the circuit model, is designed by a series of unitary gates $U(\boldsymbol{\phi})$, which is applied to an initial register of qubits to generate the ansatz state $|0\rangle^{\otimes N} \xrightarrow{U(\boldsymbol{\phi})} |\psi(\boldsymbol{\phi})\rangle$

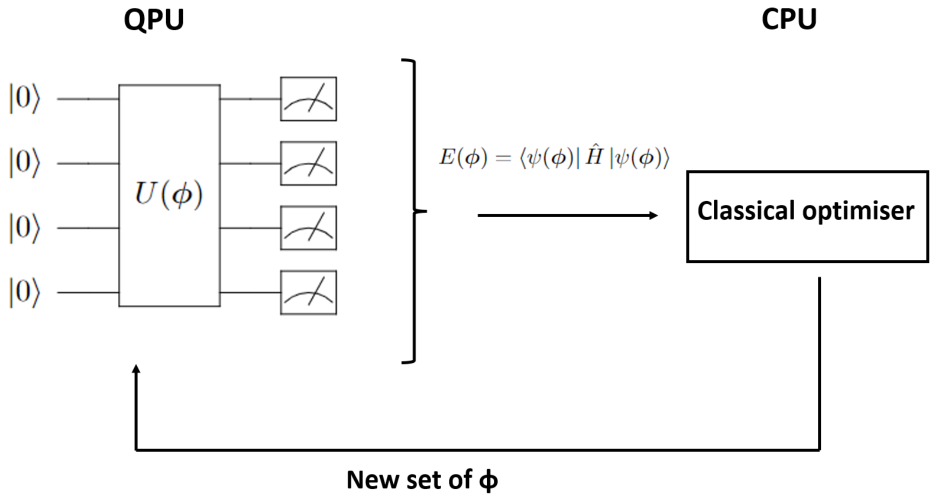


Figure 1.2: The figure illustrates the scheme of an VQE algorithm. The QPU calculates $E(\boldsymbol{\phi})$ and is passed to the CPU. The CPU subsequently updates the parameters and feed them back to the QPU. This processes is repeated until convergence of 1.19.

[11]. The ansatz state $|\psi(\boldsymbol{\phi})\rangle$ has to be sufficiently expressible such that an approximate ground state is within the parameter space, while only using a polynomial amount of optimisation parameters ϕ_i. Designing a good ansatz usually presents the difficult part of constructing a VQE problem. Furthermore, one has to represent the Hamiltonian in a suitable basis such that it can easily be measured on a quantum processor. A popular basis is the Pauli basis, which forms a group under multiplication. The Pauli group for a single qubit is given by:

$$\mathcal{P}_1 = \{\pm\hat{\mathbb{1}}, \pm i\hat{\mathbb{1}}, \pm\hat{X}, \pm i\hat{X}, \pm\hat{Y}, \pm i\hat{Y}, \pm\hat{Z}, \pm i\hat{Z}\}, \tag{1.20}$$

and is extended to N qubits by taking the N-fold tensor product of all elements in (1.20), $\mathcal{P}_N = \mathcal{P}_1^{\otimes N}$. Using this basis, the Hamiltonian is mapped to a linear combination of strings of Pauli operators

$$\hat{H} = \sum_i c_i \hat{P}_i, \tag{1.21}$$

where $\hat{P}_i$ is a string of Pauli operators and c_i its weight [11]. However, in order to avoid the exponentially increasing terms from the N-fold Pauli group a truncation is usually employed, which truncates (1.21) to only include a polynomial number of terms[2] [13]. For typical systems, such as electronic energies in molecules, this truncation provides a good approximation [11].

[2] If all Pauli strings were included, an exponential number of samples would be needed at each step, which is exponentially hard even for a quantum computer.

1.3 Quantum error-correcting codes

This section aims to describe the basics of quantum error-correction. It starts with presenting the main idea of error-correction and the difficulties of designing a quantum error-correcting code (QECC). The section moves on to describing a classic example, Shor's nine-qubit code, which will follow the discussions throughout the section. Finally, the general properties of a QECC is defined and translated to the stabilizer formalism and stabilizer codes.

1.3.1 Basics of error-correction

In any classical information processing system, noise is present, causing bit flips. Depending on the system this noise is more or less severe, meaning the probability p of a bit flip error to occur is either large or small. Due to the noise, a scheme to protect the information, also called *error-correcting code* is needed. The simplest classical error-correcting code is the repetition code. The repetition code encodes the bits $\overline{0}$ and $\overline{1}$ as [4]:

$$\overline{0} \longrightarrow 000, \tag{1.22}$$

$$\overline{1} \longrightarrow 111, \tag{1.23}$$

where $\overline{0} = 000$ and $\overline{1} = 111$ are referred to as the *logical 0* and *logical 1*, respectively [4]. The logical 0 and 1 are protected from bit flips through added redundancy. For example, if one bit flips in the logical 0 ($000 \rightarrow 001$), a majority voting with the remaining bits is done to recover the logical information ($001 \rightarrow 000$). This encoding fails whenever p is large enough to give a substantial probability of flipping more than one bit. However, in this case simply more redundancy can be added.

Just as classical information processing systems are subjected to noise, so are quantum information processing systems, and if not more. For these reasons, quantum error-correcting codes are needed. The classical repetition code displays some of the difficulties in constructing a QECC. First, due to the no cloning theorem [14] a QECC can not simply copy the quantum state several times to create a logical state. Second, as a qubit can be in a superposition of 0 and 1, noise can cause both bit flips and phase flips. Third, since measurements collapses any superpositions, observations of individual qubits can not be made to recover the original state. Instead, other recovery schemes and logical encodings have to be engineered.

1.3.2 Shor's nine-qubit code

A great example illustrating how information can be encoded and recovered without collapsing any superpositions is Shor's nine-qubit code [15]. Furthermore, it also illustrates how the continuum of single-qubit errors collapses to only bit and phase flips. Shor's nine-qubit code can correct for any single-qubit error and encodes one qubit into nine in the following manner:

$$\left|\overline{0}\right\rangle = \frac{1}{2^{3/2}}\overbrace{(|000\rangle + |111\rangle)}^{\textbf{block 1}}\overbrace{(|000\rangle + |111\rangle)}^{\textbf{block 2}}\overbrace{(|000\rangle + |111\rangle)}^{\textbf{block 3}}, \tag{1.24}$$

$$\left|\overline{1}\right\rangle = \frac{1}{2^{3/2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle), \tag{1.25}$$

where $\left|\overline{0}\right\rangle$ and $\left|\overline{1}\right\rangle$ acts as *logical 0* and *logical 1*, and maps a single qubit state $|\psi\rangle = a|0\rangle + b|1\rangle \rightarrow \left|\overline{\psi}\right\rangle = a\left|\overline{0}\right\rangle + b\left|\overline{1}\right\rangle$. If the logical encodings are subjected to a single-qubit error, we detect the error by measuring different observables with eigenvalue ±1 [16]. These observables are designed to be eigenvectors to both encodings with eigenvalue +1, and thus a measurement yielding -1 indicates an error is present. The result of the measurements gives a *syndrome*. How these measurements are performed without disturbing the logically encoded state is discussed in section 1.3.5.

To determine if the single-qubit error is a bit flip on one of the qubits in the first block, we measure $\hat{Z}_1\hat{Z}_2$ and $\hat{Z}_1\hat{Z}_3$. If both measurements yield 1, no bit flip occurred in the first block. However, if one or both of the measurements yield -1, a bit flip on either qubit one, qubit two or qubit three occurred and applying $\hat{X}_i$ with $i \in \{1, 2, 3\}$ will recover the state. The same procedure is repeated for the remaining two blocks to detect any bit flips in the logical encodings.

Instead, to determine if the single-qubit error is a phase flip on any of the code qubits the following two eigenvectors are measured:

$$\hat{X}_1\hat{X}_2\hat{X}_3\hat{X}_4\hat{X}_5\hat{X}_6, \tag{1.26}$$

$$\hat{X}_1\hat{X}_2\hat{X}_3\hat{X}_7\hat{X}_8\hat{X}_9. \tag{1.27}$$

These measurements, as for the bit flip, can result in both +1, both -1, or of opposite signs. In the case of both yielding -1, the phase flip occurred in block one and $\hat{Z}_i$ with $i \in \{1, 2, 3\}$ is applied. If instead only one of the two measurements yields -1 the phase flip occurred in block two or three and $\hat{Z}_i$ with $i \in \{4, 5, 6\}$ or $i \in \{7, 8, 9\}$ is applied to recover the state.

Mentioned in the beginning of the section, the phase and bit flip are the only single-qubit errors one has to be able to correct, which is true even though there is continuous set of possible single-qubit errors [16]. This is due to the error-detection measurements, which can be seen from considering the most general error

$$\hat{E} = \alpha\hat{\mathbb{1}} + \beta\hat{X} + \gamma\hat{Y} + \eta\hat{Z}, \tag{1.28}$$

and how it transforms the logically encoded state $\left|\overline{\psi}\right\rangle$

$$\hat{E}\left|\overline{\psi}\right\rangle = \hat{E}(a\left|\overline{0}\right\rangle + b\left|\overline{1}\right\rangle) = \alpha\hat{\mathbb{1}}\left|\overline{\psi}\right\rangle + \beta\hat{X}\left|\overline{\psi}\right\rangle + \gamma\hat{Y}\left|\overline{\psi}\right\rangle + \eta\hat{Z}\left|\overline{\psi}\right\rangle. \tag{1.29}$$

Performing the error-detection measurements the state in (1.29) will collapse to the original state $\left|\overline{\psi}\right\rangle$ times one of the Pauli operators with probabilities $\{|\alpha|^2, |\beta|^2, |\gamma|^2, |\eta|^2\}$, respectively [17]. Thus, the state collapses to either no error, a bit flip, a phase flip or both a bit flip and phase flip[3].

In general, a QECC like Shor's nine-qubit code works by first choosing a *logical encoding*. As the logical encoding is subjected to noise, an error arises in the form of a bit flip, phase flip or both. To detect the error, a *syndrome measurement* is performed, which is a measurement of simultaneous eigenvectors to the logical encodings. If an error is detected, *recovery* is performed. In the next subsection we will present the general properties of any quantum code.

1.3.3 General properties of quantum codes

The codespace $\mathcal{T}$ of a QECC is a subspace of the full Hilbert space $\mathcal{H}_N$ of dimension 2^N. The codespace contains the logical encodings of the QECC, also called codewords, which are redundantly encoded in N qubits. For instance, in Shor's code the codespace is of dimension 2 since it contains two codewords, and they are encoded in nine qubits (i.e. $\mathcal{H}_N$ is of dimension 2^9). Determining which errors the QECC can correct for, only a basis of errors need to be considered, as any linear combination of correctable errors are also correctable [17]. The basis commonly used, just as for VQE, is the Pauli basis and is given in (1.20).

To be able to correct for a set of errors $\mathcal{E} = \{\hat{E}_i\}$, the QECC must fulfil a criterion commonly known as the *Knill-Laflamme* criterion [18]

$$\langle\psi_j| \hat{E}_\alpha^\dagger \hat{E}_\beta |\psi_i\rangle = \delta_{\alpha\beta}\delta_{ji} \quad \forall \hat{E}_\alpha, \hat{E}_\beta \in \mathcal{E}, \tag{1.30}$$

where $|\psi\rangle_j, |\psi\rangle_j \in \mathcal{T}$ are the codewords of the QECC. Equation (1.30) actually encodes two separate criteria, and it is worth highlighting both of these for intuition. The first criterion is stated with the equation [17]:

$$\langle\psi_j| \hat{E}_\alpha^\dagger \hat{E}_\beta |\psi_i\rangle = 0 \quad \forall \hat{E}_\alpha, \hat{E}_\beta \in \mathcal{E}, \tag{1.31}$$

which expresses that we cannot mistake one error $\hat{E}_\alpha$ for another $\hat{E}_\beta$, by enforcing that $\hat{E}_\alpha$ and $\hat{E}_\beta$ must produce orthogonal states when acting on different codewords [17]. The second criterion relates to the syndrome measurements, which enforces that no information can be gained about the state when these are performed, i.e. the coefficients a and b in (1.29). Otherwise, possible superpositions between codewords would collapse, which is formulated with the equation [17]:

$$\langle\psi_i| \hat{E}_\alpha^\dagger \hat{E}_\beta |\psi_i\rangle = \langle\psi_j| \hat{E}_\alpha^\dagger \hat{E}_\beta |\psi_j\rangle . \tag{1.32}$$

An important variable for determining the error-correcting capability of a QECC is the distance d, which is given by the weight of the smallest operator in $\mathcal{E}$, with the weight of an operator $\hat{E}_\alpha$ defined by the number of Pauli operators in $\hat{E}_\alpha$ different from identity. For a QECC to be able to correct for t errors it needs a distance of at least $d = 2t + 1$. However, if a QECC is to only detect s errors or correct l located errors a distance of $d = s + 1$ and $d = l + 1$ is needed, respectively. For example, Shor's code can correct for any single qubit error, and thus has a distance $d = 3$.

[3] Note that $\hat{Y} = i\hat{X}\hat{Z}$, and is thus equivalent to a phase flip and bit flip error

1.3.4 Quantum error-correction in the stabilizer formalism

The stabilizer formalism offers an alternative approach to describing a large set of quantum states by leveraging group theory. Instead of describing a quantum state through its state vector, the stabilizer formalism describes the state by an Abelian subgroup $\mathcal{S}$ of $\mathcal{P}$ called the *stabilizers*, which the state in question is an eigenvector to [4]. This allows, for instance, to describe a QECC by a group of commuting operators instead of the state vectors to the codewords. Applying the stabilizer formalism is analogous to going from the Schrödinger picture to the Heisenberg picture in quantum mechanics [19].

A stabilizer code hosting c codewords in a Hilbert space H_N of dimension 2^N is defined by a stabilizer group with $N-c$ generators $\mathcal{S} = \langle \hat{g}_1, ..., \hat{g}_{N-c} \rangle$. The code space $\mathcal{T}$ is the set of vectors invariant under $\mathcal{S}$:

$$\mathcal{T} = \{ |\psi\rangle \quad | \quad \hat{g} |\psi\rangle = |\psi\rangle \quad \forall \hat{g} \in \mathcal{S} \}. \tag{1.33}$$

Since $\mathcal{S}$ is a subgroup of $\mathcal{P}$, all stabilizers either commute or anticommute with $\mathcal{P}$. Those errors $\mathcal{E} \in \mathcal{P}$ which anitcommutes with at least one stabilizer generator are correctable[4]. For instance, assume $\hat{g}_k \in \mathcal{S}$, $|\psi\rangle_i$, $|\psi\rangle_j \in \mathcal{T}$ and $\{\hat{g}_k, \hat{E}\} = 0$, then

$$\langle \psi_i | \hat{E} | \psi_j \rangle = \langle \psi_i | \hat{E} \hat{g}_k | \psi_j \rangle = - \langle \psi_i | \hat{g}_k \hat{E} | \psi_j \rangle = - \langle \psi_i | \hat{E} | \psi_j \rangle = 0, \tag{1.34}$$

and both the criterion in (1.31) and in (1.32) are fulfilled. Hence, we see if an error anticommutes with one stabilizer generator it is indeed correctable. More formally, a stabilizer code can correct all errors in some set $\mathcal{E}$ which fulfil:

$$\hat{E} = \hat{E}_\alpha^\dagger \hat{E}_\beta \in \mathcal{S} \cup \mathcal{P} \setminus \mathcal{C}(\mathcal{S}) \quad \forall \hat{E}_\alpha, \hat{E}_\beta \in \mathcal{E}, \tag{1.35}$$

where $\mathcal{C}(\mathcal{S})$ is the *centraliser* of $\mathcal{S}$ and is defined as all operators in $\mathcal{P}$ that commutes with $\mathcal{S}$: $C(\mathcal{S}) = \{ \hat{p} \in \mathcal{P} \quad | \quad \hat{p}\hat{g}_k = \hat{g}_k \hat{p} \quad \forall \hat{g}_k \in \mathcal{S} \}$. With the centraliser defined, the distance of a stabilizer code is given by the minimum weight of all operators in $\mathcal{C}(\mathcal{S}) \setminus \mathcal{S}$.

While operators in $\mathcal{S}$ fixes $\mathcal{T}$, operators in $\mathcal{C}(\mathcal{S}) \setminus \mathcal{S}$ map states in $\mathcal{T}$ between each other and thus acts as logical operators [17]. For a code with c codewords there are c independent logical operators, $\overline{Z}_i$ and $\overline{X}_i$, which generate $\mathcal{C}(\mathcal{S}) \setminus \mathcal{S}$ and follow the usual commutation rules:

$$[\overline{X}_i, \overline{X}_j] = 0 \quad \forall i, j, \tag{1.36}$$

$$[\overline{Z}_i, \overline{X}_j] = 2i\delta_{ij}\overline{Y}_i \quad \forall i, j, \tag{1.37}$$

$$[\overline{Z}_i, \overline{Z}_j] = 0 \quad \forall i, j, \tag{1.38}$$

$$\{\overline{X}_i, \overline{Z}_j\} = 0 \quad \forall i, j. \tag{1.39}$$

Concluding, we note that Shor's code is a stabilizer code. The operators we measured to check weather a phase flip or bit flip occurred are in fact the stabilizer generators of the code and are shown in table 1.1. From the generators, it is straightforward to show that $\hat{Z}_1 \hat{Z}_4 \hat{Z}_7 \in \mathcal{C}(\mathcal{S}) \setminus \mathcal{S}$, which is the operator of smallest weight in the centraliser, and thus the code is indeed of distance $d = 3$.

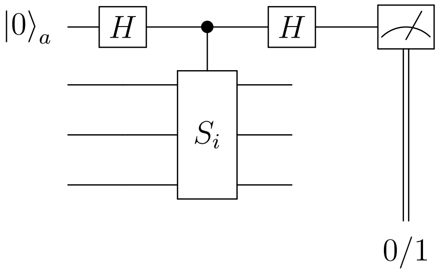
[4] Strictly speaking, $\hat{E} \in \mathcal{S}$ is also correctable, but it acts as the identity on the encoded information.

Table 1.1: The stabilizer generators of Shor's code.

$\hat{g}_1$	$\hat{X}_1$	$\hat{X}_2$	$\hat{X}_3$	$\hat{X}_4$	$\hat{X}_5$	$\hat{X}_6$	$\hat{\mathbb{1}}_7$	$\hat{\mathbb{1}}_8$	$\hat{\mathbb{1}}_9$
$\hat{g}_2$	$\hat{X}_1$	$\hat{X}_2$	$\hat{X}_3$	$\hat{\mathbb{1}}_4$	$\hat{\mathbb{1}}_5$	$\hat{\mathbb{1}}_6$	$\hat{X}_7$	$\hat{X}_8$	$\hat{X}_9$
$\hat{g}_3$	$\hat{Z}_1$	$\hat{Z}_2$	$\hat{\mathbb{1}}_3$	$\hat{\mathbb{1}}_4$	$\hat{\mathbb{1}}_5$	$\hat{\mathbb{1}}_6$	$\hat{\mathbb{1}}_7$	$\hat{\mathbb{1}}_8$	$\hat{\mathbb{1}}_9$
$\hat{g}_4$	$\hat{Z}_1$	$\hat{\mathbb{1}}_2$	$\hat{Z}_3$	$\hat{\mathbb{1}}_4$	$\hat{\mathbb{1}}_5$	$\hat{\mathbb{1}}_6$	$\hat{\mathbb{1}}_7$	$\hat{\mathbb{1}}_8$	$\hat{\mathbb{1}}_9$
$\hat{g}_5$	$\hat{\mathbb{1}}_1$	$\hat{\mathbb{1}}_2$	$\hat{\mathbb{1}}_3$	$\hat{Z}_4$	$\hat{Z}_5$	$\hat{\mathbb{1}}_6$	$\hat{\mathbb{1}}_7$	$\hat{\mathbb{1}}_8$	$\hat{\mathbb{1}}_9$
$\hat{g}_6$	$\hat{\mathbb{1}}_1$	$\hat{\mathbb{1}}_2$	$\hat{\mathbb{1}}_3$	$\hat{Z}_4$	$\hat{\mathbb{1}}_5$	$\hat{Z}_6$	$\hat{\mathbb{1}}_7$	$\hat{\mathbb{1}}_8$	$\hat{\mathbb{1}}_9$
$\hat{g}_7$	$\hat{\mathbb{1}}_1$	$\hat{\mathbb{1}}_2$	$\hat{\mathbb{1}}_3$	$\hat{\mathbb{1}}_4$	$\hat{\mathbb{1}}_5$	$\hat{\mathbb{1}}_6$	$\hat{Z}_7$	$\hat{Z}_8$	$\hat{\mathbb{1}}_9$
$\hat{g}_8$	$\hat{\mathbb{1}}_1$	$\hat{\mathbb{1}}_2$	$\hat{\mathbb{1}}_3$	$\hat{\mathbb{1}}_4$	$\hat{\mathbb{1}}_5$	$\hat{\mathbb{1}}_6$	$\hat{Z}_7$	$\hat{\mathbb{1}}_8$	$\hat{Z}_9$

1.3.5 Measuring stabilizers

We here present how to measure a stabilizer without disturbing the encoded information, which generalises to any observable with eigenvalues ±1. To measure a stabilizer a projective measurement onto its eigenspace is performed [4]. Such a measurement is achieved using an ancilla qubit as illustrated in the circuit below:



To see the circuit indeed measures the stabilizer $\hat{S}_i$, we follow the dynamics of the circuit until the measurement

$$\begin{aligned}\hat{H}_a\hat{U}^{S_i}\hat{H}_a\left|0\right\rangle_a\left|\psi\right\rangle &= \frac{\hat{H}_a}{\sqrt{2}}(\left|0\right\rangle_a\left|\psi\right\rangle + \left|1\right\rangle_a\hat{S}_i\left|\psi\right\rangle) \\ &= \frac{1}{2}(1+S_i)\left|0\right\rangle_a\left|\psi\right\rangle + \frac{1}{2}(1-S_i)\left|1\right\rangle_a\left|\psi\right\rangle,\end{aligned} \tag{1.40}$$

where $\hat{U}^{S_i}$ is a control operator applying $\hat{S}_i$ if the control qubit is one, and S_i is the eigenvalue of $\hat{S}_i$. Thus, if $|\psi\rangle$ is in the eigenstate with eigenvalue 1 of $\hat{S}_i$, the measurement of the ancilla qubit will result in $|0\rangle_a$ with unit probability. If instead $|\psi\rangle$ is in the eigenstate with eigenvalue -1 of $\hat{S}_i$, the measurement outcome of the ancilla qubit will be $|1\rangle_a$.

Chapter 2

Graph states and graph codes

With the stabilizer formalism introduced, we proceed with discussing graph states and graph codes, which offers a graphical description to stabilizer states and stabilizer codes [20, 21]. The graph state is one of the fundamental building blocks of this thesis, as it will be used in all applications we consider. We begin the chapter by introducing graph states and an operation called local complementation, which allows to identify graph states equivalent under local (single-qubit) operations and can be described by an intuitive graph operation. We end the chapter by introducing graph codes, where we discuss how a code is built, how information is encoded, and present an example.

2.1 Graph states and local complementation

A graph state $|G\rangle$ is a quantum state defined by N-qubits initialised in the plus eigenstate of the Pauli-X operator and a pattern E of control phases [22]:

$$|G\rangle = \prod_{\{\alpha,\beta\}\in E} \hat{U}^{CZ}_{\alpha\beta} \left|+\right\rangle^{\otimes N}, \tag{2.1}$$

where $\hat{U}^{CZ}$ is given by:

$$\hat{U}^{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{2.2}$$

As the name indicates, a graph state is associated to a graph object $G(V, E)$ where qubits are represented by a set of vertices and the pattern of $\hat{U}^{CZ}$ by a set of edges. This offers a convenient way of illustrating graph states. As an example, the graph object of the star graph state containing four qubits entangled with three $\hat{U}^{CZ}$ is illustrated in figure 2.1.

Graph states are a stabilizer states, and can thus be uniquely described by their stabilizer

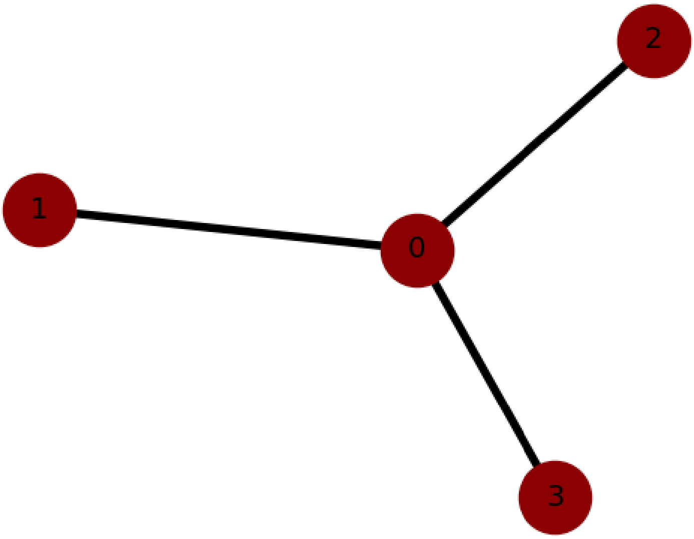


Figure 2.1: The figure illustrates the star graph state with four qubits (vertices) entangled with three control phases (edges).

generators [20]. A graph state with N-qubits have N-stabilizer generators given by:

$$\{\hat{g}_n\} = \{\hat{X}_n \prod_{i \in N_G(n)} \hat{Z}_i\} \quad \text{for} \quad n = 1, 2, .., N, \tag{2.3}$$

where $N_G(n)$ is defined as the neighbourhood of qubit n, which is the set of all its neighbouring qubits $\{j \quad | \{j, n\} \in E\}$. Here $\hat{X}_n$ represents the Pauli-X operator on qubit n and $\hat{Z}_i$ the Pauli-Z operator on qubit i. As we proceed, we will denote a generator's origin from the position of its Pauli-X operator, i.e. $\hat{g}_n$ has position n.

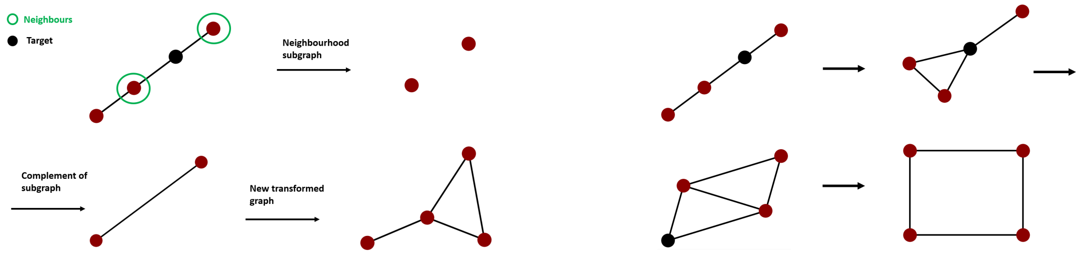
Another important unitary group when discussing graph states is the local Clifford group $\mathcal{C}$. The local Clifford group is defined by the set of local unitaries $\{\hat{U}_i\}$ fulfilling: $\mathcal{P} = \hat{U}_i \mathcal{P} \hat{U}_i^\dagger$, which entails a stabilizer state is mapped to a stabilizer state under the action of $\hat{U}_i$ [22]. The single qubit Clifford group $\mathcal{C}_1 = \{\hat{H}, \hat{S}\}$ is generated by the Hadamard operator given in (1.3) and the phase operator

$$\hat{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \tag{2.4}$$

and the N-fold tensor product of $\mathcal{C}_1$ generates the N-qubit local clifford group $\mathcal{C}_N = \mathcal{C}_1^{\otimes N}$. Two graph states $|G\rangle$ and $|G'\rangle$ are said to be local clifford equivalent (LC-equivalent) if they can be transformed into each other via an operation in $\mathcal{C}_N$. An important property for graph states is that two graph states are LC-equivalent iff they are connected by a sequence of local complemetation operations [22]. Local complemetation is a particular type of local-Clifford operation, which when applied on qubit α is described by the unitaries

$$\hat{U}_\alpha^{LC} = \sqrt{-i\hat{X}_\alpha} \prod_{\{\beta \in N_G(\alpha)\}} \sqrt{i\hat{Z}_\beta} = e^{-i\frac{\pi}{4}\hat{X}_\alpha} \prod_{\{\beta \in N_G(\alpha)\}} e^{i\frac{\pi}{4}\hat{Z}_\beta}, \tag{2.5}$$

which has a simple interpretation in term of graph transformations. Local complementation applied to vertex α of the graph $G(V, E)$ transforms its subgraph $G[N_G(\alpha)] \subseteq G(V, E)$



(a) (b)

Figure 2.2: (a) Graphically illustrates local complementation with the black vertex as the target qubit (α). If there are no edge connecting two vertices of the neighbourhood subgraph, the complement will add one. However, if there is an edge, the complement will remove it. Furthermore, (b) illustrates the full local equivalence class associated with the four qubit linear graph, excluding isomorphic graphs.

to its complement [23] and is illustrated in figure 2.2, along with *local equivalence class*[1] associated with the four qubit linear graph. The connection between such graph transformation rules and LC-equivalent graph states represents a very interesting and important link that allows to study quantum states just in terms of their graph representation. This property makes graph states a very convenient tool to investigate quantum codes, which is the main reason we focus on them here.

As a final remark on local complementation, the table 2.1 presents how the Pauli operators of the respective qubits in the graph state transform under conjugation with $\hat{U}_\alpha^{LC}$. With $|G'\rangle = \hat{U}_\alpha^{LC} |G\rangle$, the transformation ensures the stabilizer generators of $|G\rangle$ transforms to the stabilizer generators of $|G'\rangle$ [24].

Table 2.1: The transformation of the Pauli operators of each qubit in the graph state under conjugation with U_α^{LC}.

qubit	X	Y	Z
α	X	-Z	Y
$N_G(\alpha)$	-Y	X	Z
$V \setminus (N_G(\alpha) \cup \alpha)$	X	Y	Z

2.2 Graph codes

Having introduced graph states, we proceed the discussion by introducing graph codes. Just as graph states are stabilizer states, graph codes are stabilizer codes [21]. The section starts with defining the logical encodings and their operators, and then follows up by describing how a physical qubit can be encoded a graph code.

[1] The *local equivalence class* associated with a graph, is the class of graphs accessible from local complementation only.

2.2.1 Code construction

A graph code can be defined with the graph state $|G\rangle$ and a subset of qubits I [25]. The subset I is necessary to define the logical operators and to encode a physical qubit in the graph code. Given I and $|G\rangle$, the logical encodings are defined as:

$$\left|\bar{0}\right\rangle = |G\rangle\,, \tag{2.6}$$

$$\left|\bar{1}\right\rangle = \prod_{n\in I} \hat{Z}_n \,|G\rangle\,. \tag{2.7}$$

Using the fact that $\hat{Z}_n$ commutes with all $\hat{U}^{CZ}_{ij}$ and that $\hat{U}^{CZ}_{ij}$ is unitary we can prove that $\left|\bar{0}\right\rangle$ and $\left|\bar{1}\right\rangle$ are orthogonal:

$$\langle\bar{0}|\bar{1}\rangle = \langle +|^{\otimes N} \hat{U}^{\dagger} \prod_{n\in I} \hat{Z}_n \hat{U} \,|+\rangle^{\otimes N} = \langle +|^{\otimes N} \,|+\rangle^{\otimes N-|I|} \,|-\rangle^{\otimes |I|} = 0, \tag{2.8}$$

where $\hat{U}$ denotes the collection of control phase operators constructing the graph state.

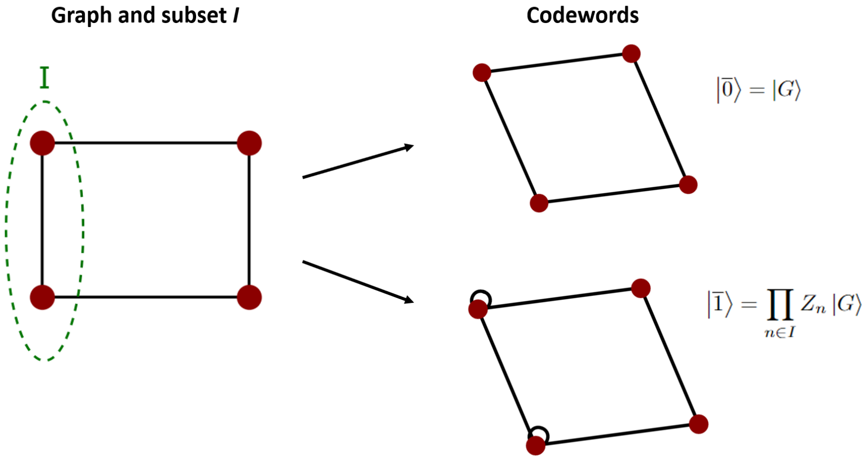

Graph and subset ***I***
Codewords
I
$\left|\bar{0}\right\rangle = |G\rangle$
$\left|\bar{1}\right\rangle = \prod_{n\in I} Z_n \,|G\rangle$

Figure 2.3: The left picture shows the code graph and the subset *I*. The two pictures to the right displays the respective graph objects of the codewords. The logical zero is the original graph and logical one is the original graph with self-loops on the vertices in the subset *I*.

Moreover, logical one can be viewed as the original graph (i.e. logical zero) but with self-loops as depicted in figure 2.3, where self-loops represents local Pauli-Z operations instead of control phases [25]. Logical one share most of its generators with the original graph state, but with generators originating from the subset I modified as

$$\hat{g}_n = \begin{cases} -\hat{g}_n, & \text{if } n \in I. \\ \hat{g}_n, & \text{otherwise.} \end{cases} \tag{2.9}$$

With the logical encodings the logical operators are defined as:

$$\overline{X} = \prod_{n\in I} \hat{Z}_n, \tag{2.10}$$

$$\overline{Z} = \hat{X}_n \prod_{i\in N_G(n)} \hat{Z}_i, \quad n \in I. \tag{2.11}$$

Note that there is freedom in choosing $\overline{Z}$, any generator originating from I suffice. Also, note a logical operator times a stabilizer is still a logical operator $\langle \overline{X}, \overline{Z}\rangle \times \hat{S}$. Furthermore, showing $\overline{X}$ and $\overline{Z}$ operates on the logical encodings in the correct way is straightforward:

$$\overline{X}\left|\overline{0}\right\rangle = \prod_{n\in I} \hat{Z}_n \left|\overline{0}\right\rangle = \prod_{n\in I} \hat{Z}_n \left|G\right\rangle = \left|\overline{1}\right\rangle, \tag{2.12}$$

$$\overline{X}\left|\overline{1}\right\rangle = \prod_{n\in I} (\hat{Z}_n)^2 \left|\overline{0}\right\rangle = \left|\overline{0}\right\rangle, \tag{2.13}$$

$$\overline{Z}\left|\overline{0}\right\rangle = \hat{X}_n \prod_{i\in N_G(n)} \hat{Z}_i \left|\overline{0}\right\rangle = \hat{g}_n \left|\overline{0}\right\rangle = \left|\overline{0}\right\rangle, \tag{2.14}$$

$$\overline{Z}\left|\overline{1}\right\rangle = -(-\hat{g}_n)\left|\overline{1}\right\rangle = -\left|\overline{1}\right\rangle. \tag{2.15}$$

Given the logical encodings and their respective generators, a graph code with two codewords for a N-qubit graph can be constructed. The code's *N-1* generators are found from (2.3) and are given by:

$$\hat{g}'_n = \begin{cases} \hat{g}'_n = \overline{Z}\hat{g}_n, & \text{if } n \in I. \\ \hat{g}'_n = \hat{g}_n, & \text{otherwise.} \end{cases} \tag{2.16}$$

That is, the generators of the code $\{\hat{g}'_n\}$ are the generators of logical zero in (2.3), with the generators originating from the subset I modified as: $\hat{g}_n \longrightarrow \overline{Z}\hat{g}_n$. This ensures the correct commutation relations between the logical operators and the generators[2].

2.2.2 Encoding information into graph codes

There are two ways of encoding the state of a generic qubit $|\psi\rangle_p = a|0\rangle + b|1\rangle$ into a graph code in a measurement based manner [25]. In the first approach, the graph state $|G\rangle_N$ (N-qubits) is first prepared and then the physical qubit is entangled by performing control phases with the qubits in the subset I. This creates the composite state

$$\begin{aligned} a|0\rangle_p \left|\overline{0}\right\rangle_N + b|1\rangle_p \left|\overline{1}\right\rangle_N &= |+\rangle_p (a\left|\overline{0}\right\rangle_N + b\left|\overline{1}\right\rangle_N) + |-\rangle_p (a\left|\overline{0}\right\rangle_N - b\left|\overline{1}\right\rangle_N) \\ &= |+\rangle_p \left|\overline{\psi}\right\rangle_N + |-\rangle\, \overline{Z}\left|\overline{\psi}\right\rangle_N, \end{aligned} \tag{2.17}$$

[2] Remember the logical operators are in the centraliser of the code and thus commute with all generators.

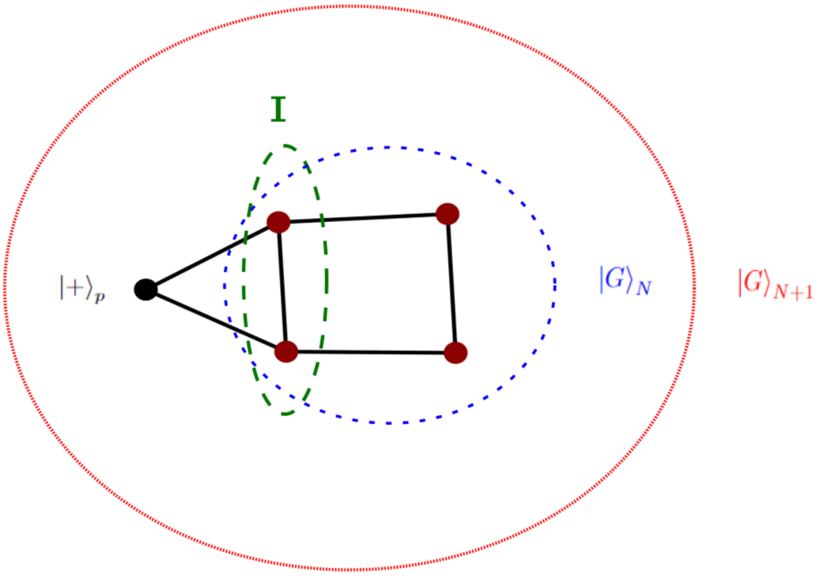


Figure 2.4: The figure shows the full graph $|G\rangle_{N+1}$ in red, the code graph $|G\rangle_N$ in blue and the subset I in green. Furthermore, the physical qubit to be encoded is represented as the black vertex.

where p refers to the physical qubit whose state is to be encoded, and $\left|\bar{0}\right\rangle_N$ and $\left|\bar{1}\right\rangle_N$ are the logical encodings of the graph code defined by $|G\rangle_N$ and I. Measuring the physical qubit in the Pauli-X basis projects its state to the graph code up to a logical-Z operation.

The second approach is realised by first preparing the graph state $|G\rangle_{N+1}$, with the physical qubit first prepared in $|+\rangle$, which is displayed in figure 2.4. Then, measuring the physical qubit in the basis $\{|\psi\rangle, |\psi^\perp\rangle\}$ prepares the code in the measurement outcome up to conjugation. To see how this is done, we first rewrite $|G\rangle_{N+1}$ as:

$$\begin{aligned} |G\rangle_{N+1} &= \prod_{j\in I} \hat{U}_{pj}^{CZ} \, |+\rangle_p \, |G\rangle_N \\ &= \frac{|0\rangle_p \, |G\rangle_N + |1\rangle_p \prod_{j\in I} \hat{Z}_j \, |G\rangle_N}{\sqrt{2}} \\ &= \frac{|0\rangle_p \left|\bar{0}\right\rangle_N + |1\rangle_p \left|\bar{1}\right\rangle_N}{\sqrt{2}}. \end{aligned} \tag{2.18}$$

Again, p symbolizes the physical qubit, and $\left|\bar{0}\right\rangle_N$ and $\left|\bar{1}\right\rangle_N$ the logical encodings of the graph code defined by $|G\rangle_N$ and I. Moving to the more general basis

$$|\psi\rangle = \sqrt{\alpha}\,|0\rangle + \sqrt{1-\alpha}e^{i\xi}\,|1\rangle\,, \tag{2.19}$$

$$\left|\psi^\perp\right\rangle = \sqrt{1-\alpha}e^{-i\xi}\,|0\rangle - \sqrt{\alpha}\,|1\rangle\,, \tag{2.20}$$

where $\alpha \in [0,1]$ and $\xi \in [0, 2\pi]$, we can rewrite (2.18)

$$\begin{aligned}|G\rangle_{N+1} &= \frac{(\sqrt{\alpha}\,|\psi\rangle_p + \sqrt{1-\alpha}e^{i\xi}\,|\psi^\perp\rangle_p)\,|\overline{0}\rangle_N + (\sqrt{1-\alpha}e^{-i\xi}\,|\psi\rangle_p - \sqrt{\alpha}\,|\psi^\perp\rangle_p)\,|\overline{1}\rangle_N}{\sqrt{2}} \\ &= \frac{|\psi^\perp\rangle_p\,(\sqrt{1-\alpha}e^{i\xi}\,|\overline{0}\rangle_N - \sqrt{\alpha}\,|\overline{1}\rangle_N) + |\psi\rangle_p\,(\sqrt{\alpha}\,|\overline{0}\rangle_N + \sqrt{1-\alpha}e^{-i\xi}\,|\overline{1}\rangle_N)}{\sqrt{2}} \\ &= \frac{|\psi^\perp\rangle_p\left|\overline{\psi^{*\perp}}\right\rangle_N + |\psi\rangle_p\left|\overline{\psi^*}\right\rangle_N}{\sqrt{2}}.\end{aligned} \tag{2.21}$$

Hence, by measuring the physical qubit in the general basis (2.19) and (2.20), its state is projected to the logical encodings of the N-qubit graph code up to conjugation.

2.2.3 Distance three graph code

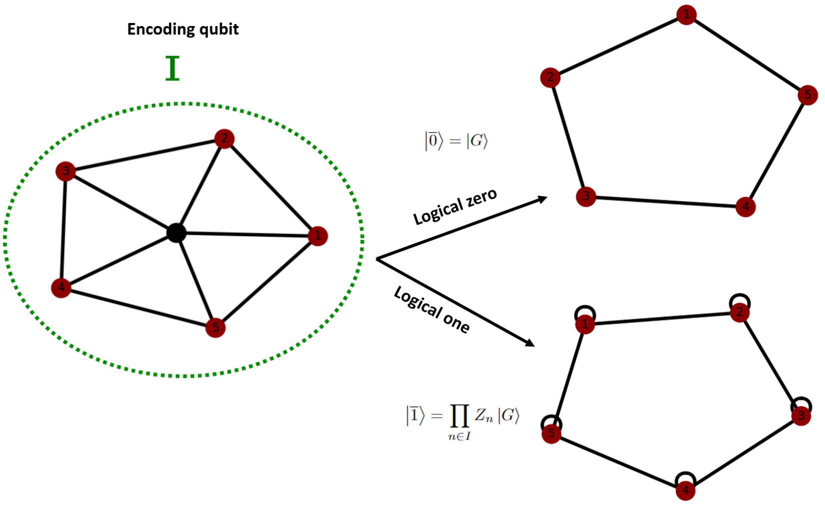

Encoding qubit
I
$|\overline{0}\rangle = |G\rangle$
Logical zero
Logical one
$|\overline{1}\rangle = \prod_{n\in I} Z_n\,|G\rangle$

Figure 2.5: To the left, the figure illustrates how a physical qubit (black) is encoded in the pentagon graph code. To the right, the codewords and their graph objects are shown.

We here describe an example of a graph code: the pentagon code. The pentagon graph state with all of its qubits as the subset I defines the smallest graph code with distance $d = 3$. The codewords of the graph code, along with the encoding of a physical qubit is illustrated in figure 2.5. The logical operators and the stabilizer generators are given by:

$$\begin{aligned}\overline{X} &= \hat{Z}_1\hat{Z}_2\hat{Z}_3\hat{Z}_4\hat{Z}_5, \\ \overline{Z} &= \hat{Z}_1\hat{X}_2\hat{Z}_3\hat{\mathbb{1}}_4\hat{\mathbb{1}}_5, \\ \mathcal{S} = \{\hat{g}_1 &= \hat{Y}_1\hat{Y}_2\hat{Z}_3\hat{\mathbb{1}}_4\hat{Z}_5, \\ \hat{g}_2 &= \hat{Z}_1\hat{Y}_2\hat{Y}_3\hat{Z}_4\hat{\mathbb{1}}_5, \\ \hat{g}_3 &= \hat{Z}_1\hat{X}_2\hat{\mathbb{1}}_3\hat{X}_4\hat{Z}_5, \\ \hat{g}_4 &= \hat{\mathbb{1}}_1\hat{X}_2\hat{Z}_3\hat{Z}_4\hat{X}_5\}.\end{aligned}$$

To verify that the code is indeed $d = 3$, one has to check that all single-qubit Pauli operators and two-qubit Pauli operators anticommute with at least one of the stabilizer generators. We note that there is at least two different generators which have two different Pauli operators on the same qubit. Hence, all single and two-qubit Pauli operators anticommute with at least one generator. However, this is not the case for all three-qubit Pauli operators. For example, consider $\hat{E} = \hat{Y}_1\hat{Y}_2\hat{Y}_3\hat{\mathbb{1}}_4\hat{\mathbb{1}}_5$, it commutes with all generators

$$[\hat{g}_1, \hat{E}] = [\hat{g}_2, \hat{E}] = [\hat{g}_3, \hat{E}] = [\hat{g}_4, \hat{E}] = 0, \tag{2.22}$$

and thus belongs to the centraliser of the graph code, giving the code a distance of three. Another way of seeing this, is that there is no logical operator of weight less than three. Thus, any single-qubit error will map the logical encodings out of the code space, which will be detected measuring the stabilizers.

Chapter 3

A quantum dot graph state generator

In this chapter we discuss a physical system which is emerging as a promising platform for the generation of the graph states described in the previous chapter: the spin-photon interface in a quantum dot (QD). In this thesis we will focus on this platform, which is sometimes called the "photonic machine gun" [26] due to the capability of the quantum dot to generate entangled single photons on demand in multiple rounds. The chapter starts with a general introduction to the physical system, and we will then describe a protocol for generating entangled states of time-bin encoded photonic qubits with a single quantum dot. We continue describing the most relevant noise sources in this physical system and a model to numerically simulate them. Finally, we report a search for all graph states that can be generated with this system.

3.1 The quantum dot

Quantum dots are emerging as an excellent photon source for quantum computing and quantum information due to three main properties: (1) they can deterministically emit single photons, (2) coupling between their internal spin state and optical transitions form a platform for a spin-photon interface, (3) it forms a solid-state system which is easy to integrate into nanophotonic structures and waveguides. In particular, we will here focus on QDs embedded in a photonic crystal waveguide (PWC). Excellent coupling to the guiding waveguide, high-efficiency collection of the emitted photons, and manipulation of the QD decay rates have all already been demonstrated to be possible with this platform [27, 28].

Quantum dots are fabricated by growing one semiconducting-material inside a different semiconducting-material. Due to the different band gaps of the semiconducting materials, a three-dimensional quantum well is created, which quantizes the electronic-structure of the valence and conduction bands to discrete energy levels, much like that of an atom[1] [29, 30]. In this thesis, we are considering a InAs/GaAs QD, where InAs (indium arsenide) is grown inside GaAs (gallium arsenide).

[1] Even though the QD consist of over thousands of atoms.

In a neutral QD, optically exciting an electron to the conduction band subsequently creates a hole in the valence band. The electron and hole forms a bound state called an exciton and as the exciton decays, a photon is emitted with energy dependent on the band gap and geometry of the QD [30]. While the neutral QD allows for single photon generation, it cannot contribute as a matter qubit as it decays through dipole radiation in the order of nanoseconds. However, by charging the QD with an external bias voltage and applying a magnetic field, a matter qubit can be encoded in the spin of the charge carrier and coupling between the spin and the emitted photons is possible [30]. The QD can be both positively or negatively charged, with the ground state having an electron in conduction band or a hole in the valence band, respectively. We will work with a positively charged QD, whose excited state contains two holes and one electron and is called a *trion*.

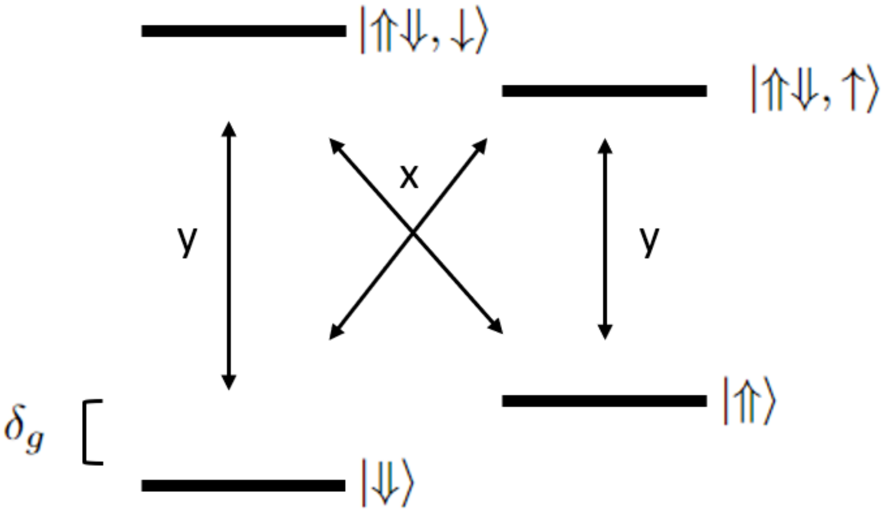

$|\Uparrow\Downarrow,\downarrow\rangle$
$|\Uparrow\Downarrow,\uparrow\rangle$
x
y
y
δ_g
$|\Uparrow\rangle$
$|\Downarrow\rangle$

Figure 3.1: The figure illustrates the electronic level structure of a positively charged QD, with x and y indicating the polarisation of the different trion dipoles. The hole spin is denoted $\{|\Uparrow\rangle, |\Downarrow\rangle\}$, and the spin of the electron denoted as $\{|\uparrow\rangle, |\downarrow\rangle\}$. Furthermore, δ_g is the energy splitting between the ground states $\{|\Uparrow\rangle, |\Downarrow\rangle\}$.

Figure 3.1 illustrates the electronic level structure of a positively charged QD, which is a four level system. As explained above, applying an in-plane magnetic field splits the ground state degeneracy to two stable ground states $\{|\Uparrow\rangle, |\Downarrow\rangle\}$, separated by δ_g. The excited state levels $\{\{|\Uparrow\Downarrow\rangle, \uparrow, \{|\Uparrow\Downarrow, \downarrow\rangle\}$ consist of an electron excited to the conduction band and two holes in a singlet state, with the spin of the electron denoted $\{|\uparrow\rangle, |\downarrow\rangle\}$. In this work, we will encode the positively charged QD energy levels as:

$$|\Downarrow\rangle \rightarrow |0\rangle \tag{3.1}$$

$$|\Uparrow\rangle \rightarrow |1\rangle \tag{3.2}$$

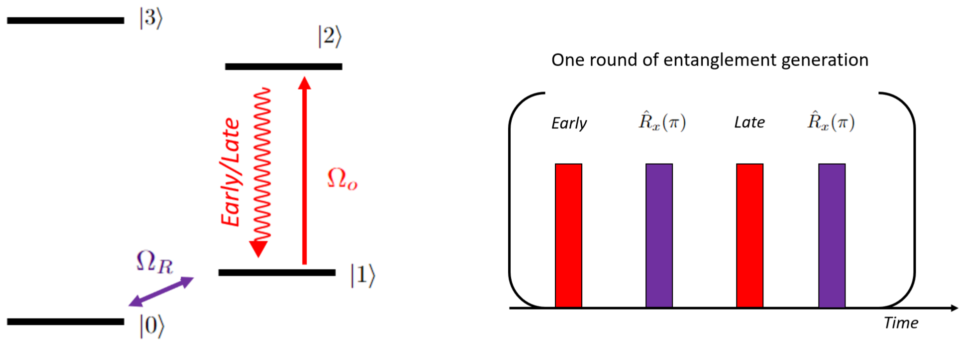
$$|\Uparrow\Downarrow, \uparrow\rangle \rightarrow |2\rangle \tag{3.3}$$

$$|\Uparrow\Downarrow, \downarrow\rangle \rightarrow |3\rangle\,. \tag{3.4}$$

To summarise QDs are excellent single-photon sources as they can be integrated into nanophotonic structures. The neutral QD is not suitable as a qubit as it decays through dipole radiation in the order of nanoseconds. However, if the QD is charged by an external bias voltage a matter qubit can be encoded in the spin of the charge carrier. With the charged qubit, coupling is possible between the spin and the emitted photons, enabling deterministic entanglement generation, as we will see in the next section.

3.2 Entanglement generation in a spin-photon interface

We describe here a protocol for the generation of time-bin photons, entangled with the spin of the hole in a positively charged QD. The protocol to realise the time-bin entanglement generation is related to the Lindner-Rudolph protocol [26], and is described in great detail in Ref. [30]. The ideal excitation and decay scheme of the protocol, along with the spin rotations can be depicted in figure 3.2. In the protocol, two lasers are used: one (purple colour in 3.2) resonant with the spin $|0\rangle \longleftrightarrow |1\rangle$ transition for spin manipulation, and one (red colour in figure 3.2) resonant with the $|1\rangle \rightarrow |2\rangle$ transition. The x-dipole ($|0\rangle \longleftrightarrow |2\rangle$) is suppressed by the nanophotonic structure.



(a)

(b)

Figure 3.2: The time-bin entanglement protocol (a) illustrates the energy levels and transitions of the protcol, with Ω_o as the Rabi frequency of the excitation laser and Ω_R the Rabi frequency of the rotation laser. (b) Illustrates the optical and spin pulse sequences for one full round of the protocol.

For simplicity, let us start describing the generation of a single time-bin photon entangled with the spin state. The spin-photon state is initialised in $|\Psi\rangle = (a\,|0\rangle_S + b\,|1\rangle_S)\,|\emptyset\rangle_p$, where $|0\rangle_S$ and $|1\rangle_S$ are the QD states in figure 3.2 and $|\emptyset\rangle_p$ indicates no photons in the optical mode. The protocol now follows:

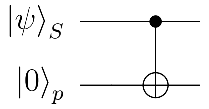
1. Optically excite the $|1\rangle_S \longrightarrow |2\rangle_S$ transition, generating an *early* photon $a_e^\dagger\,|\emptyset\rangle_p = |e\rangle_p$.
2. Flip the ground states of the spin with a $R_x(\pi)$ pulse, $|0\rangle_S \longrightarrow |1\rangle_S$ and $|1\rangle_S \longrightarrow |0\rangle_S$.

3. Optically excite the $|1\rangle_S \longrightarrow |2\rangle_S$ transition again, generating a *late* photon $a_l^\dagger |\emptyset\rangle_p = |l\rangle_p$.
4. Do a final $R_x(\pi)$ pulse, flipping the ground states again.

With the encoding for the time bin qubit, $|l\rangle_p = |0\rangle_p$ and $|e\rangle_p = |1\rangle_p$, implies the final state is written as $|\Psi\rangle = a|0\rangle_S |0\rangle_p + b|1\rangle_S |1\rangle_p$. For clarity, we follow the evolution of the protocol:

$$\begin{aligned} a|0\emptyset\rangle + b|1\emptyset\rangle &\xrightarrow{1} a|0\emptyset\rangle + b|1e\rangle \\ \xrightarrow{2} a|1\emptyset\rangle + b|0e\rangle &\xrightarrow{3} a|1l\rangle + b|0e\rangle \\ \xrightarrow{4} +a|0l\rangle + b|1e\rangle &= a|00\rangle + b|11\rangle\,. \end{aligned} \tag{3.5}$$

The protocol can be seen as initialising the photonic qubit state in $|0\rangle_p$ and performing a controlled-not gate (CNOT) with the spin, as illustrated in the circuit below:



Thus, the protocol is easily generalised to generate multi-qubit entangled states since each entanglement generation simply amounts to performing a CNOT between a newly generated photonic qubit and the spin qubit. For instance, initialising the spin in $|\psi\rangle_S = |+\rangle$ (i.e. $a = b = 1/\sqrt{2}$) and performing the above protocol N-times generates the N-qubit Greenberger–Horne–Zeilinger (GHZ) state [31]: $\frac{|0\rangle^{\otimes N} + |1\rangle^{\otimes N}}{\sqrt{2}}$. While the protocol is interesting and can in principle generate a large number of entangled photons from a single QD, in practice there are various noises which need to be taken into account to investigate its performance in real implementations. This will be discussed in the coming section.

3.3 Error model

In this section we will discuss how realistic imperfections affect the above protocol and how these can be modelled in simulations. The error model generalises that of Ref. [30] to multi-photon entangled states, with also some additional slight modifications. The model is based on a Monte-Carlo approach using collapse operators with probability distributions determined by experimental parameters. Given a quantum state to be simulated, e.g. a graph state, its density matrix is constructed by simulating N Monte-Carlo trajectories and taking the average as:

$$\hat{\rho} = \frac{\sum_{j=1}^{N} |\psi\rangle_j \langle\psi|_j}{N}. \tag{3.6}$$

The simulations consider a Hilbert space constructed as

$$\mathcal{H} = \{1,2,3,4\} \otimes (\{\emptyset, e\} \otimes \{\emptyset, l\})^{\otimes n}, \tag{3.7}$$

where $\{1, 2, 3, 4\}$ are the QD energy levels, $\emptyset$ an optical mode in the vacuum state, e/l early/late photon, and n the number of generated photons.

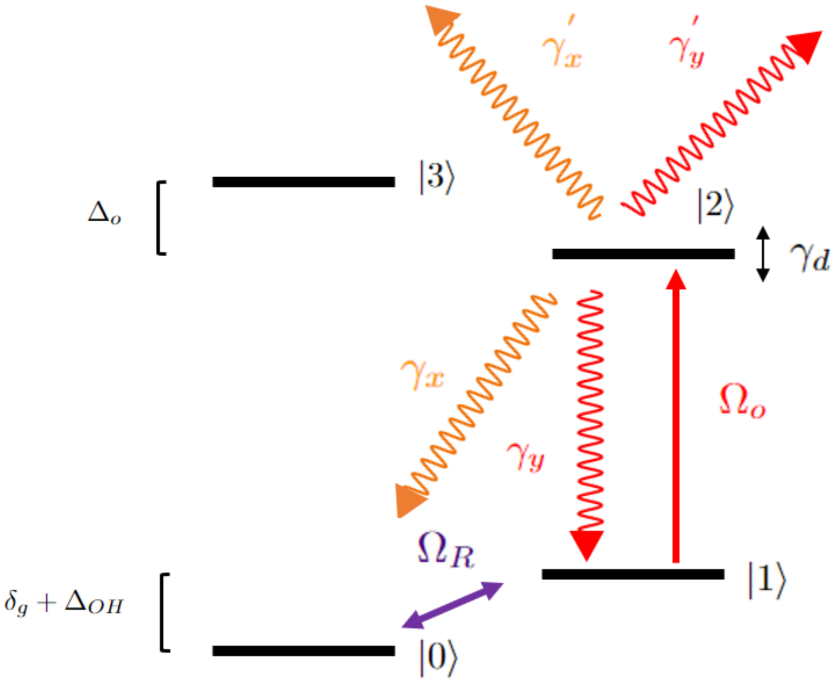


Figure 3.3: Errors which can arise during the time-bin entanglement protocol. First, due to the finite detuning $\delta_e = \Delta_o + \Delta_{OH} + \delta_g$, there is a probability of driving the transition $|0\rangle \longrightarrow |3\rangle$. Second, the Overhauser field causes an inhomogenous broadening of the ground state splitting by an amount Δ_{OH}. Third, due to finite cyclicity the QD can decay through the x-dipole transition. Fourth, photons can be lost, which is indicated by primes. Finally, while excited, the QD can be subjected to pure dephasing γ_d through phonon scattering.

3.3.1 Rotation errors

There are mainly two mechanisms giving rise to spin rotations errors: The Overhauser field and the laser induced spin flips. As discussed in section 3.1 the QD, even though behaving as an atom, is in fact composed by several thousands of atoms. These atoms nuclei all have a nuclear spin (In=9/2, Ga=3/2 and As =3/2), which couples to the charge carrier spin through the hyperfine interaction. This will perturb the energy splitting δ_g of the ground states (see figure 3.3) by an amount Δ_{OH}. Because of the inhomogenous broadening[2] of Δ_{OH}, the spin will decohere at a time scale defined by the spin coherence time $T_2^* = \frac{\sqrt{2}}{\sigma_{OH}}$, which is dependent on the standard deviation σ_{OH} of Δ_{OH}. A thorough derivation of this decoherence and how it affects the rotation pulses can be found in Ref. [30]. In principle, the entanglement protocol is insensitive to the ground state dephasing caused by the Overhauser field [32], which is due to the first $\hat{R}_x(\pi)$ in figure 3.2 flipping

[2] It is inhomogenous since Δ_{OH} changes between experiments.

the ground states acting as an Echo pulse [30].

Although the entanglement protocol is, in principle, insensitive to the ground state dephasing caused by the Overhauser field, it does perturb the rotation pulses. The ground state perturbation Δ_{OH} perturbs the detuning between the sidebands of the Raman pulses, which affect the rotation pulse fidelity[3]. Taking the ground state perturbation Δ_{OH} into account, the perturbed rotation operator takes the form:

$$\hat{U}_{rot}(\theta_r, \phi_a) = \cos(\theta_r)\hat{\mathbb{1}} - i\sin(\theta_r)\frac{\Omega_r(\cos(\phi_a)\hat{X} + \sin(\phi_a)\hat{Y}) - (\Delta_D + \Delta_{OH})\hat{Z}}{\sqrt{\Omega_r^2 + (\Delta_D + \Delta_{OH})^2}}, \tag{3.8}$$

where Ω_r is the Rabi frequency of the rotation laser and Δ_D is the detuning between the sidebands of the Raman pulse and the ground state splitting. The angle $\theta_r = \frac{T_r}{2}\sqrt{\Omega_r^2 + (\Delta_D + \Delta_{OH})^2}$ defines the rotation angle with T_r being the pulse duration, and ϕ_a defines the azimutal angle of rotation. During simulations, a new Δ_{OH} is drawn from a normal distribution $\mathcal{N}(0, \sqrt{2}/T_2^*)$ for each trajectory, which accounts for its inhomogenous distribution.

While the Overhauser field is quite well understood as well as how it affects the rotation fidelity, it is not the main source of error during rotations. The main source of error seems to be a laser induced spin flips. The rate of the spin flips scales with the laser power, but the physics behind it is not yet fully understood [30]. A hypothesis mentioned in Ref. [30] is that the rotation laser is producing photocreation, which creates random holes that can be seen as an effective spin flip. Although the physics is not yet fully understood, this error can be modelled by a spin flip rate parameter κ.

During simulation of rotation pulses, we wish to apply (3.8) for a certain pulse duration T_r. To include the spin flips, we introduce the collapse operators:

$$\hat{F}_1 = \sqrt{\kappa}\,|0\rangle\langle 1|\,, \tag{3.9}$$

$$\hat{F}_2 = \sqrt{\kappa}\,|1\rangle\langle 0|\,. \tag{3.10}$$

By drawing t from a Poisson distribution

$$P(t) = \kappa e^{-\kappa t}, \tag{3.11}$$

we evolve $|\psi\rangle$ with (3.8) for a pulse duration of t and then apply one of the spin flip operators (3.9) and (3.10) with probability,

$$p_j = \frac{\langle\psi|\,\hat{F}_j^\dagger\hat{F}_j\,|\psi\rangle}{\sum_i\langle\psi|\,\hat{F}_i^\dagger\hat{F}_i\,|\psi\rangle}, \tag{3.12}$$

which collapses $|\psi\rangle$ as:

$$|\psi\rangle \rightarrow \frac{\hat{F}_j\,|\psi\rangle}{\sqrt{p_j}}. \tag{3.13}$$

This procedure is repeated by drawing a new t from (3.11) until we reach the intended pulse duration T_r and the rotation is finished.

[3] For more information on Raman pulses see Refs. [33, 30].

3.3.2 Excitation and decay errors

During excitation and decay of the QD there are several errors which can occur. These are illustrated in figure 3.3 and will be simulated with collapse operators similar to the spin flips. During excitation of the QD there are mainly two sources of errors. First, the optical driving laser Ω_o, which is set to drive the transition $|1\rangle \longrightarrow |2\rangle$, has a finite detuning $\delta_e = \Delta_o + \delta_g + \Delta_{OH}$ to the $|3\rangle$ level, and thus there is a finite probability to drive the $|0\rangle \longrightarrow |3\rangle$ transition. Second, as the excitation pulse is of finite duration, the QD may decay during the pulse emitting a pulse photon. This leads to the possibility of emitting two photons during one round of the entanglement protocol[4].

While we have to account for both of these events during simulations (detuned excitation and pulse photons), we assume that the off-resonant decay photons and the pulse photons can be perfectly filtered out. This is possible as the off-resonant photons will be of a different frequency and the pulse photons will have a different emission time and different bandwidth. Assuming perfect filtering of pulse photons is in contrast to Ref. [30], but it allow us to ignore them in the construction of the Hilbert-space, which subsequently alleviates the size of the Hilbert-space needed for the simulations. In this way, we can simulate larger states with more photons.

The ideal excitation operator and the collapse operators, which takes the discussed infidelities into account are defined as:

$$\hat{E}_1 = c_{00} |0\rangle \langle 0| + c_{11} |1\rangle \langle 1| + c_{30} |3\rangle \langle 0| + c_{21} |2\rangle \langle 1| \quad \text{(ideal)}, \tag{3.14}$$

$$\hat{E}_2 = c'_{00} |0\rangle \langle 0| \quad \text{(excite-decay)}, \tag{3.15}$$

$$\hat{E}_3 = c'_{11} |1\rangle \langle 1| \quad \text{(excite-decay)}, \tag{3.16}$$

$$\hat{E}_4 = c'_{10} |1\rangle \langle 0| \quad \text{(excite-decay)}, \tag{3.17}$$

$$\hat{E}_5 = c'_{01} |0\rangle \langle 1| \quad \text{(excite-decay)}, \tag{3.18}$$

$$\hat{E}_6 = c'_{21} |2\rangle \langle 1| \quad \text{(excite-decay-excite)}, \tag{3.19}$$

$$\hat{E}_7 = c'_{30} |3\rangle \langle 0| \quad \text{(excite-decay-excite)}, \tag{3.20}$$

$$\hat{E}_8 = c'_{20} |2\rangle \langle 0| \quad \text{(excite-decay-excite)}, \tag{3.21}$$

$$\hat{E}_9 = c'_{31} |3\rangle \langle 1| \quad \text{(excite-decay-excite)}. \tag{3.22}$$

Operator $\hat{E}_1$ represents the ideal evolution during excitation. Operators $\hat{E}_2$ to $\hat{E}_5$ represents a process where the QD is excited and decays with a pulse-photon. Finally, operators $\hat{E}_6$ to $\hat{E}_9$ represent a process where the QD is first excited, then decays with a pulse-photon and finally is re-excited again. Here, for the probability coefficients in (3.14-3.22) we will consider the values estimated in Ref. [30].

During decay, the photon can either decay through the ideal decay path, through the suppressed x-dipole transition, through the correct y-dipole transition and be lost, or it can decay through the ideal decay path with the QD subjected to phonon scattering

[4] In an ideal round of the entanglement generation only one photon is emitted, but it exists in both early and late time-bin.

during excitation. An important parameter, which characterises the ratio between the y-dipole and x-dipole transitions is the cyclicity and is given by:

$$C = \frac{\gamma_y + \gamma'_y}{\gamma_x + \gamma'_x}, \tag{3.23}$$

with γ_y and γ_x representing the decay rate of the y-transition and x-transition into the waveguide mode and γ'_y and γ'_x representing the decay rate of the y-transition and x-transition into a non-guided mode (i.e. photon lost). The ratio between the decay rate into the waveguided modes non-guided modes is defined by the beta factor β, which can be engineered close to unity in PCW [34].

Mentioned above, while the QD is excited it is subjected to phonon scattering, which can be defined by a pure dephasing rate γ_d. Phonon scattering will kill the coherence of the QD by inducing a random phase to the state. Experimentally, this is seen by a reduced indistinguishability of the emitted photons [30]

$$I = \frac{\gamma}{\gamma + 2\gamma_d}, \tag{3.24}$$

where γ is the total decay rate of the QD. Given the cyclicity, indistinguishability and the external efficiency η (such as detection and collection efficiency), the ideal operator and the collapse operators during decay are defined as:

$$\hat{D}_1 = |0\rangle\langle 0| + |1\rangle\langle 1| + \sqrt{\frac{I\eta C}{1+C}}a^\dagger_{e,l}|2\rangle\langle 0|, \tag{3.25}$$

$$\hat{D}_2 = \sqrt{\frac{C}{1+C}}|3\rangle\langle 0|, \tag{3.26}$$

$$\hat{D}_3 = \sqrt{\frac{1}{1+C}}|2\rangle\langle 1|, \tag{3.27}$$

$$\hat{D}_4 = \sqrt{\frac{1}{1+C}}|3\rangle\langle 1|, \tag{3.28}$$

$$\hat{D}_5 = \sqrt{\frac{(1-\eta)C}{1+C}}|2\rangle\langle 0|, \tag{3.29}$$

$$\hat{D}_6 = \sqrt{\frac{(1-I)\eta C}{1+C}}a^\dagger_{e,l}|2\rangle\langle 0|. \tag{3.30}$$

Here, $a^\dagger_{e,l}$ is the creation operator, with e and l indicating early and late photons, respectively. Operator $\hat{D}_1$ represents the ideal decay path, while $\hat{D}_6$ represents the ideal decay path with the QD subjected to phonon scattering. Further, operators $\hat{D}_2$ to $\hat{D}_5$ represent a decay path where the photon is lost due to filtering or inefficiencies in detection/collection (characterised by η). In this work we set $\eta = 1$, and focus entirely on the intrinsic infidelities of the QD system.

The evolution of the quantum state $|\psi\rangle$ during excitation and decay follows by applying one of the respective collapse or the ideal operators

$$|\psi\rangle \to \frac{\hat{O}_j|\psi\rangle}{\sqrt{p_j}}, \tag{3.31}$$

where $\hat{O}_j$ represents one operator from (3.14-3.22) during excitation or one from (3.25-3.30) during decay. Furthermore, the probability p_j of applying operator $\hat{O}_j$ is given by:

$$p_j = \frac{\langle\psi| \hat{O}_j^\dagger \hat{O}_j |\psi\rangle}{\sum_i \langle\psi| \hat{O}_i^\dagger \hat{O}_i |\psi\rangle}. \tag{3.32}$$

3.3.3 Measurement errors

Measuring a photon in the Z-basis simply amounts to detecting the photon and recording its time-bin. However, to measure a time-bin photon in the XY-plane, the late and early time-bins need to be interfered [30]. In practice, this is achieved via a Franson interferometer sending the early component into a delay arm equal to the time separation of the time-bins, and then interfering early with the late component on a 50:50 beam splitter. A schematic of such an interferometer is illustrated in figure 3.5 and it transforms the late and early modes to the detector modes as:

$$\begin{pmatrix} a_1^\dagger \\ a_2^\dagger \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} -ie^{-i\omega} & 1 \\ e^{-i\omega} & -i \end{pmatrix} \begin{pmatrix} a_e^\dagger \\ a_l^\dagger \end{pmatrix}. \tag{3.33}$$

Here, $a_{1/2}^\dagger$ are the creation operators of respective detector (D1/D2) in the interferometer. Furthermore, ω is the phase acquired from the delay arm, which determines the azimuthal angle on the Bloch sphere, and the $i's$ are from reflecting off the beam splitter [30]. Detection in one of the detectors then maps to measuring one of the two states of an orthogonal basis in the XY-plane. Defining $\zeta = w + \frac{\pi}{2}$ and with the time-bin encoding defined in section 3.3, the detectors perform a measurement in the $M(\zeta) = \{\frac{|0\rangle + e^{-i\zeta}|1\rangle}{\sqrt{2}}, \frac{|0\rangle - e^{-i\zeta}|1\rangle}{\sqrt{2}}\}$ basis.

Errors arising in the measurement of photons will be modelled by slight rotations of the measurement basis. Given we want to perform a measurement in the basis $M(\zeta)$, a slight rotation by an angle θ is applied transforming $M(\zeta) \rightarrow e^{\frac{-i\theta\hat{P}\cdot\hat{n}}{2}} M(\zeta)$. Here, $\hat{P}$ stand for the Pauli operators and $\hat{n}$ an unit vector, which is set to be orthogonal to $M(\zeta)$. This is illustrated in figure 3.5 when $M(\zeta = \pi)$ (e.g. the Y-basis) is slightly rotated around the z-axis.

3.3.4 Post selection and experimental parameters

After the density matrix has been constructed post selection is applied, which projects the density matrix to have n photons. This is performed with the following projection operator

$$\hat{P}_S = \hat{\mathbb{1}}_{QD} \otimes (|\emptyset e\rangle \langle\emptyset e| + |\emptyset l\rangle \langle\emptyset l|)^{\otimes n}, \tag{3.34}$$

which transforms the density matrix as follows

$$\hat{\rho} \rightarrow \frac{\hat{P}_S \hat{\rho} \hat{P}_S^\dagger}{tr(\hat{P}_S \rho \hat{P}_S^\dagger)}. \tag{3.35}$$

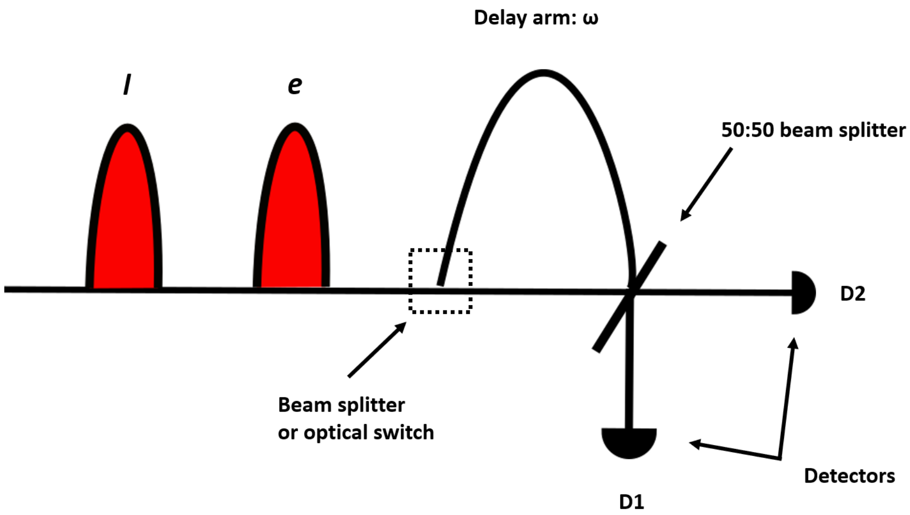


Figure 3.4: Figure illustrates the schematic of a Franson interferometer for time-bin qubit analysis. The early and late time bins are pictured as red pulses. The early component is sent to the delay arm where it acquires a phase ω before it is interfered with the late component at a 50:50 beam splitter. The two different detectors are then mapped to two orthogonal states of a basis on the XY-plane.

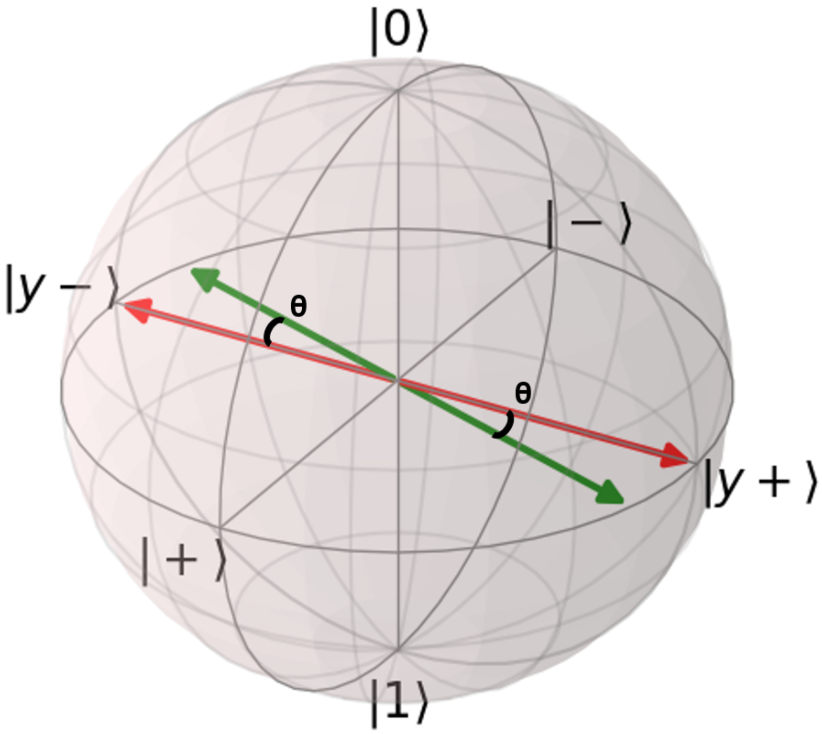


Figure 3.5: Figure illustrating errors to the Y-basis. The red coloured vectors indicate the case of no errors (i.e. $\theta = 0$), and green with errors (i.e. $\theta \neq 0$).

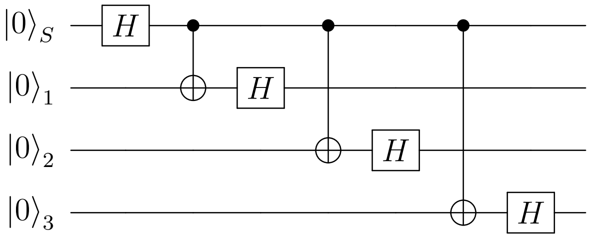
As a summary of this section, we table the realistic experimental parameters needed for the simulations [30]. This is useful as they will be continuously referred to throughout the text.

Table 3.1: Experimental parameters

Cyclicity	C = 14.7
Indistinguishability	I = 0.957
Spin flip rate	$\kappa = 0.021$ ns^{-1}
Spin coherence time	$T_2^* = 23.2$ ns
Rabi frequency	$\Omega_r = \frac{\pi}{7}$ ns^{-1}

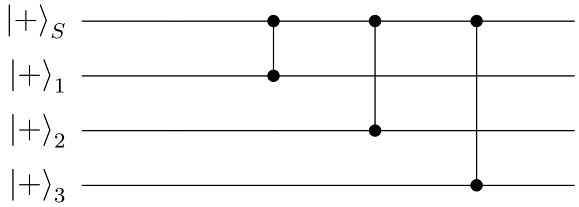
3.4 Graph state search

Having introduced the graph state in chapter 2 and explained how multi-photon entangled states can be generated by consecutive applications of CNOTs between different photonic time-bin qubits and the spin, we proceed this chapter with an exhaustive search of all possible graph states which can be generated with our time-bin entanglement protocol and local operations. The CNOT generation outlined in section 3.3, with the spin initialised in $|+\rangle_S = \frac{|0\rangle_S + |1\rangle_S}{\sqrt{2}}$ followed by a $\hat{H}$ on the generated photon is in a graphical picture equivalent to adding a leaf to the quantum dot spin qubit (i.e. a $\hat{U}^{CZ}$ between the photon and spin). This is seen from first considering the circuit below



and then using the commutation relation $\hat{U}^{CNOT} \otimes \hat{H}_{target} = \hat{H}_{target} \otimes \hat{U}^{CZ}$ and rewriting

the circuits as:



which indeed is equivalent to adding leafs to the quantum dot spin qubit. Further, we extend the class of states we can generate by considering possible local operations on the spin in between the rounds of the protocol, as well as on the photons. Because single-photon operations can be easily and deterministically performed with linear optics, this addition will still keep the protocol fully deterministic.

In our search, we will make extensive use of the very important property described in section 2.1 which states that from a given graph state we can identify all the other graph states reachable with local operations by just looking at what graphs can be obtained under local complementation (LC) operations. This enables us to perform the search by just looking at the properties of graphs instead of the quantum states they represent, making the search immensely more efficient.

Algorithm 1 Graph finder

```
Require:
  $N = 2$    ▷ Initialise number of qubits to two.
  $storing\_graphs$ = ←two_qubit_line    ▷ Dictionary for storing all generated graphs in lists of their respective local equivalence class. Initialise with the only two qubit line graph.
  while $N \leq 9$ do    ▷ Generating graphs up to 9 qubits/nodes.
      $classes\_step\_before$ ← all graphs from step before
      $storing\_classes$ ← empty list
      for $class$ in $classes\_step\_before$ do
          for $graph$ in $class$ do
              $new\_graph$ = AddNode($graph$)    ▷ Generate the new graph by adding a qubit/node to node zero.
              if $first\_graph$ is True then    ▷ If this is the first graph for this number of qubits/nodes $N$.
                  $new\_class$ = GetClass($new\_graph$)    ▷ Get the full local equivalence class for $new\_graph$.
                  $storing\_classes \leftarrow storing\_classes + new\_class$
              else
                  $LC\_equiv$ ← CheckLC($new\_graph$, $storing\_class$)    ▷ Check if $new\_graph$ already belongs to any of the local equivalence classes already generated.
                  if $LC\_equiv$ is True then
                      break
                  else
                      $new\_class$ = GetClass($new\_graph$)
                      $storing\_classes \leftarrow storing\_classes + new\_class$
                  end if
              end if
          end for
      end for
      $storing\_graphs[N] \leftarrow storing\_classes$
      $N \leftarrow N + 1$
  end while
```

The search is done numerically with a recursive algorithm. The algorithm works by at each step adding a leaf to the QD (vertex 0) for all graphs generated form the step before, and then generating all accessible LC-equivalence classes. The pseudo-code for the algorithm is outlined in Algorithm 1.

In the pseudo-code, three subroutines are used. The subroutine `AddNode` takes a graph as input and updates it by adding a leaf to the 0-th vertex. This mimics the generation of a new photon by the quantum dot.

The subroutine `CheckLC` checks if the *new graph* is locally equivalent to any of the already generated graphs. To test whether two graphs are LC-equivalent, `CheckLC` implements an algorithm originally presented in Ref. [20], which scales as $\mathcal{O}(n^4)$ with n representing the number of vertices.

The subroutine `GetClass` explores a graph's LC-equivalence class, i.e. the class of graph states that can be reached from it by just doing single-qubit operations. This part is the most computationally expense task of the search, and presents the main bottleneck of the algorithm when trying to perform the search to graph states with large numbers of qubits. In fact, finding a graphs LC-class has been shown to belong to the computational complexity class $\#\mathbb{P}$ [35]. While the search could be made more efficient by instead finding the LC-equivalence class up to isomorphism as in Ref. [36], we could potentially miss some graphs from the search as isomorphic graphs are important[5].

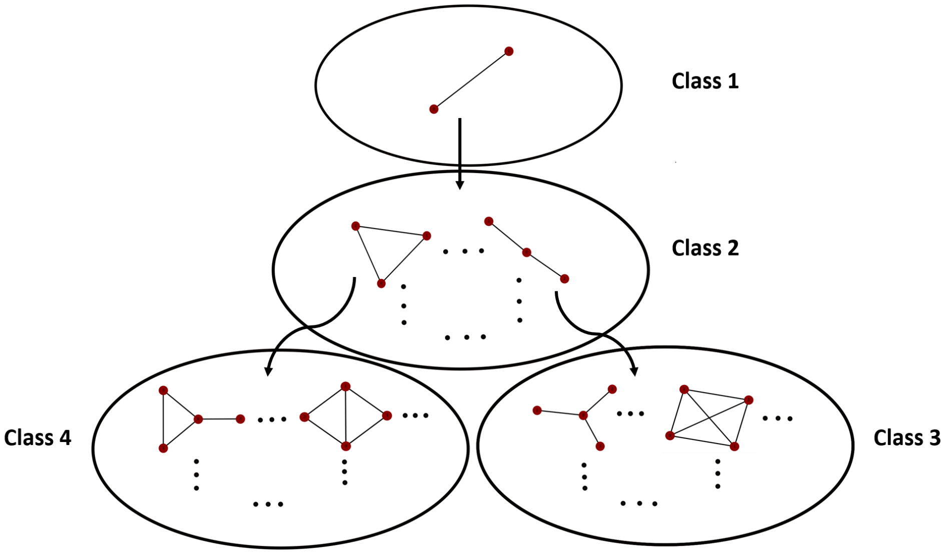


Figure 3.6: A graphical description of the first two rounds of algorithm 1. Here the circles represent a LC-equivalence class, with the dots indicating the remaining graphs of the class.

Figure 3.6 graphically displays the first two rounds of the algorithm. Along with the graphs shown in figure 3.6 the search resulted in 22,000+ graphs for up to nine vertices, which are shown in table 3.2 and displayed in Appendix A for up to five vertices. From table 3.2 we see that our graph generation protocol misses to capture significantly more graphs as the number of vertices increases, where the total number of graphs for N-vertices is given by: $N_{tot} = 2^{\binom{N}{2}}$ [22]. This is also illustrated in figure 3.7, which indicates an exponential decay in the number of accessible graph with number of vertices, similar to that shown in Ref. [37].

[5] This is because the search is dependent on the position of the QD.

Table 3.2: Number of accessible graphs compared to the total number of graphs.

Number of qubits	Number of graphs accessible	Total number of graphs
3	4	8
4	16	64
5	64	1024
6	256	32768
7	1024	2097152
8	4096	268435456
9	16384	68719476736

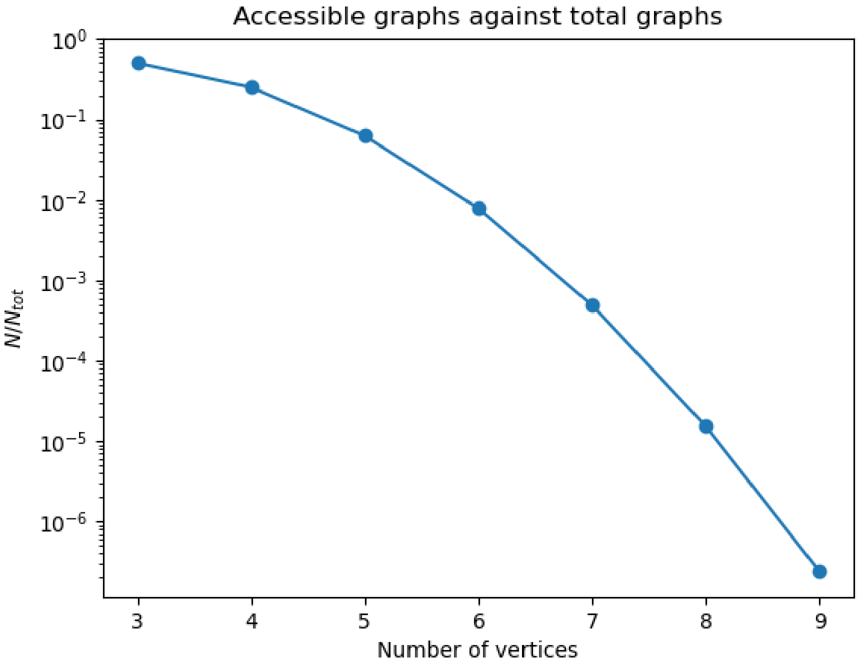


Figure 3.7: The figure displays the number of accessible graphs (N) divided by the total number of graphs (N_{tot}) for a given number of vertices.

Chapter 4

Quantum algorithms using quantum dot generated graph states

This chapter further explores the quantum algorithms presented in the introduction and presents simulations of their implementation using graph states generated from the error model described in the previous chapter. These are simulated in a NISQ fashion where errors are not corrected for, but instead kept during the computation. In this setting we would like to study the performance of the different algorithms. Grover's and Deutsch's algorithms presented here are performed through measurement based quantum computing, which is in contrast to the circuit model of computation introduced in section 1.1. Thus, the chapter starts with an introduction to this paradigm of quantum computing. Furthermore, the chapter introduces a protocol to perform VQE algorithms with our QD graph state generator, and simulates a simple two-qubit implementation.

4.1 Measurement based quantum algorithms

In quantum computing there exist two main paradigms of computing models, which are the circuit model and measurement based quantum computing (MBQC) model [38][1]. As presented in chapter 1, in the circuit model gates are applied sequentially to a quantum state and readout is performed in the computational basis ($|0\rangle, |1\rangle$). Instead of directly applying gates on qubits, MBQC mediates gates through measurements by utilising entanglement. This section aims to introduce the key features of MBQC and present results from simulating Grover's and Deutsch's algorithms with graph states generated with the error model in section 3.3.

4.1.1 Measurement based quantum computing

The key ingredient in MBQC is the family of entangled states called cluster states, which are graph states forming a lattice. By performing single-qubit measurements on part of the qubits of the cluster state gates are propagated to neighbouring qubits. Measurements,

[1] There are other paradigms as well, such as adiabatic quantum computing [39] or fusion-based quantum computing [40]

unlike unitary gates, are irreversible processes, and thus MBQC also adopts the name *one way quantum computing* [41]. To demonstrate how a measurement can propagate a gate, consider the 1D cluster state with two qubits in figure 4.1 where a measurement of the left qubit is performed in the X-basis.

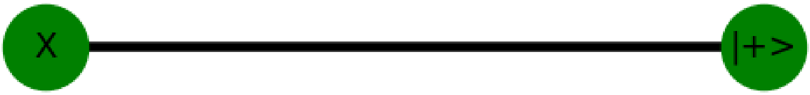


Figure 4.1: The figure display a 1D cluster state measuring the left qubit in the X-basis and leaving qubit two unmeasured. As always the line represents a $\hat{U}^{CZ}$ between the qubits.

The process in figure 4.1 amounts to applying one of the projection operators $\hat{X}_{\pm} = \frac{1}{2}(|0\rangle_1 \pm |1\rangle_1)(\langle 0|_1 \pm \langle 1|_1)$ to the cluster state $|\psi\rangle_C$. If the measurement results in X_+, we get:

$$\frac{\hat{X}_+ |\psi\rangle_C}{\sqrt{\langle\psi_C| \hat{X}_+ |\psi\rangle_C}} = \frac{1}{2}(|0\rangle_1 + |1\rangle_1)(\langle 0|_1 + \langle 1|_1) |\psi\rangle_C = |+\rangle_1 |0\rangle_2 = \hat{H}_2 |+\rangle_1 |+\rangle_2 . \tag{4.1}$$

The computation above can be seen as the 1D cluster state composing one *logical qubit* on which a Hadamard gate is applied. Further, with the following measurement bases,

$$M(Z) = \{|0\rangle , |1\rangle\}, \tag{4.2}$$

$$M(\chi) = \{M_+(\chi), M_-(\chi)\} = \{\frac{|0\rangle + e^{i\chi} |1\rangle}{\sqrt{2}}, \frac{|0\rangle - e^{i\chi} |1\rangle}{\sqrt{2}}\}, \tag{4.3}$$

any single-qubit gate can be performed on a 1D cluster state[2] [42]. This is shown in figure 4.2 for a 1D cluster state with five physical qubits composing one logical qubit. For a 1D cluster state, measuring a qubit in the general basis $M(\chi)$ with outcome $M_+(\chi)$ propagates $HR_z(\chi)$ to the next qubit in the chain. Hence, given all the measurements result in the plus state, figure 4.2 amounts to applying

$$\hat{H}\hat{H}\hat{R}_z(\gamma)\hat{H}\hat{R}_z(\beta)\hat{H}\hat{R}_z(\alpha) = \hat{R}_z(\gamma)\hat{R}_x(\beta)\hat{R}_z(\alpha), \tag{4.4}$$

to the final qubit. Equation 4.4 is an Euler rotation with Euler angles α, β and γ [3], and thus gives access to any single-qubit gate. Finally, with the bases in (4.2) and (4.3) and with a 2D cluster state single-qubit gates, along with two-qubit gates can be performed. Thus, the 2D cluster state is considered a universal resource for MBQC[3] [44].

An important detail missing from the above discussion is that measurement outcomes are random, and so far we have only considered when the measurement results in $M_+(\chi)$. If instead a measurement in the cluster state in figure 4.2 results in $M_-(\chi)$, the gate

[2] Measuring a qubit in the basis $M(Z)$ simply removes it form the cluster.

[3] For MBQC, the 2D-cluster state is the analogue to the universal set of gates in the circuit model of computing [5]. Furthermore, with a 3D cluster state, fault tolerance can be achieved [43].

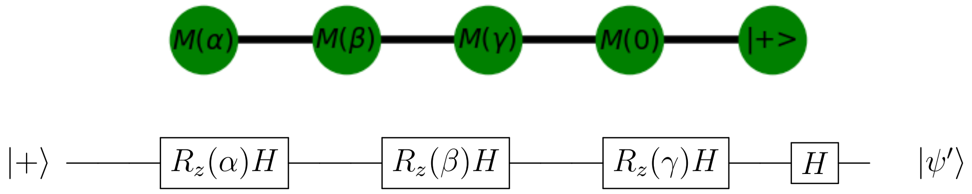


Figure 4.2: On top, the figure illustrates a 1D cluster state with five qubits, where the first four qubits are being measured in the bases $M(\alpha)$, $M(\beta)$, $M(\gamma)$ and $M(0)$, respectively. Qubit five is the final qubit of the encoding and is left unmeasured, indicated by $|+\rangle$. Below is the corresponding circuit of the measurement pattern with the measurements yielding M_+ for all $\{\alpha, \beta, \gamma, 0\}$.

$\hat{X}\hat{H}\hat{R}_z(\chi)$ is propagated to the next qubit. Here, the extra $\hat{X}$ is denoted a *Pauli by-product*. Thus, measuring in the $M(\chi)$ basis amounts to applying $\hat{X}^a\hat{H}\hat{R}_z(\chi)$ to the next qubit, with $a = 0$ and $a = 1$ for $M_+(\chi)$ and $M_-(\chi)$ outcome, respectively. Taking the randomness of measurement outcomes into account the operations applied to the output qubit from the measurement sequence in figure 4.2 are instead

$$\hat{X}^{a_4}\hat{H}\hat{X}^{a_3}\hat{H}\hat{R}_z(\gamma)\hat{X}^{a_2}\hat{H}\hat{R}_z(\beta)\hat{X}^{a_1}\hat{H}\hat{R}_z(\alpha) = \tag{4.5}$$

$$\hat{X}^{a_4}\hat{Z}^{a_3}\hat{X}^{a_2}\hat{Z}^{a_1}\hat{R}_z(-\gamma)\hat{R}_x(-\beta)\hat{R}_z(\alpha), \tag{4.6}$$

where a_i stands for the measurement outcome of qubit i. Furthermore, we have used the following commutation relations:

$$\hat{X}\hat{R}_z(-\gamma) = \hat{R}_z(\gamma)\hat{X}, \tag{4.7}$$

$$\hat{Z}\hat{R}_x(-\beta) = \hat{R}_x(\beta)\hat{Z}. \tag{4.8}$$

Equation 4.5 is not the rotation we intended to apply, which is due to the Pauli by-products. To correct this, we have to adopt adaptive measurements, where the measurement of a given qubit depend on the outcomes of the earlier measurements. In the case of figure 4.2, we choose to measure qubit 2 and 3 in $M((-1)^{a_1}\beta)$ and $M((-1)^{a_2}\gamma)$, respectively. This corresponds to applying

$$\hat{X}^{a_4}\hat{H}\hat{X}^{a_3}\hat{H}\hat{R}_z((-1)^{a_2}\gamma)\hat{X}^{a_2}\hat{H}\hat{R}_z((-1)^{a_1}\beta)\hat{X}^{a_1}\hat{H}\hat{R}_z(\alpha) = \tag{4.9}$$

$$\hat{X}^{a_4}\hat{Z}^{a_3}\hat{X}^{a_2}\hat{Z}^{a_1}\hat{R}_z(\gamma)\hat{R}_x(\beta)\hat{R}_z(\alpha), \tag{4.10}$$

to the final output qubit, which is what we intended up to the Pauli by-products. With all the by-products commuted to the left, they can be dealt with by reinterpreting the measurement outcome of the final output qubit. For example, if $\hat{U}|\psi\rangle = c|+\rangle + d|-\rangle$ and the by-product is $\hat{Z}$, then the by-product maps the final output state to $\hat{X}\hat{U}|\psi\rangle = c|-\rangle + d|+\rangle$. However, this can be corrected by simply be reinterpreting the measurement outcome s as $s \oplus a_1$, with a_1 being the Pauli-Z exponent [42]. Hence, in a measurement based computation, two forms of adaptive operations have to be performed [45]:

1. *Adaptive measurements* as measurements depend on the outcome of previous measurements.

2. *Adaptive correction* to the final output qubit as Pauli by-products have been accumulated during the computation.

4.1.2 Measurement based Grover's algorithm

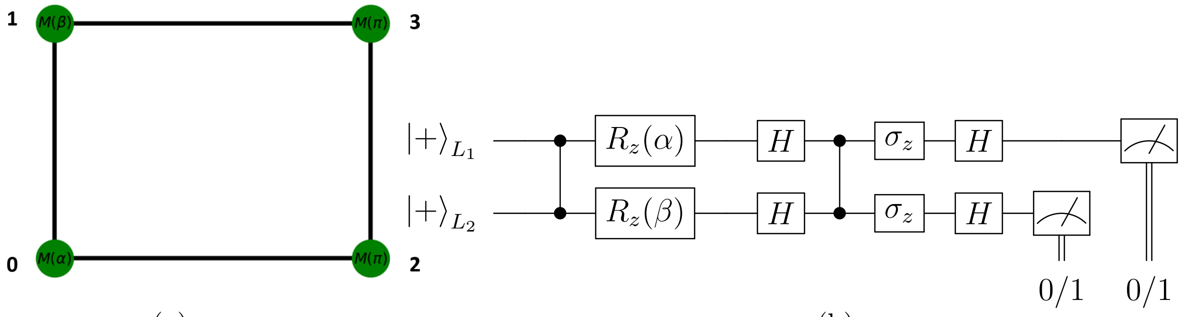
The four qubit 2D cluster state shown in figure 4.3a, denoted the box graph, will act as our resource to perform Grover's search algorithm with two input qubits. That is, the four qubit graph hosts two logical qubits: $\left|\psi_{N=4}^{Box}\right\rangle = \left|+\right\rangle_{L_1}\left|+\right\rangle_{L_2}$, with qubit one and three composing L_1 and qubit zero and two composing L_2. The procedure of the algorithm follow Ref. [45] and is depicted in figures 4.3a and 4.3b. The outline is as follows:

(i) The two logical qubits are prepared in a superposition of the logical encodings, $\left|0\right\rangle_{L_1}\left|0\right\rangle_{L_2}, \left|0\right\rangle_{L_1}\left|1\right\rangle_{L_2}, \left|1\right\rangle_{L_1}\left|0\right\rangle_{L_2}, \left|1\right\rangle_{L_1}\left|1\right\rangle_{L_2}$.

(ii) The black box performs the tagging operation, which requires a $\hat{U}^{CZ}$ between the logical qubits, followed by $\hat{R}_{z1}(\alpha)$ and $\hat{R}_{z2}(\beta)$, with the rotation angles $\{\alpha,\ \beta\}$ determining the tagged item as:

- $\{0{,}0\} \longrightarrow \left|1\right\rangle_{L_1}\left|1\right\rangle_{L_2}$
- $\{0{,}\pi\} \longrightarrow \left|1\right\rangle_{L_1}\left|0\right\rangle_{L_2}$
- $\{\pi{,}0\} \longrightarrow \left|0\right\rangle_{L_1}\left|1\right\rangle_{L_2}$
- $\{\pi,\pi\} \longrightarrow \left|0\right\rangle_{L_1}\left|0\right\rangle_{L_2}$

(iii) The black box performs the inversion around the mean, which requires $\hat{H}$, followed by a $\hat{U}^{CZ}$ and $\hat{H}\hat{Z}$ on both L_1 and L_2. Subsequently, measurements are performed in the computational basis (see figure 4.3b).



(a) (b)

Figure 4.3: (a) The box graph with labels indicating the physical qubits and their corresponding measurement basis for performing Grover's algorithm. (b) The quantum circuit version of Grover's algorithm, with L_1 and L_2 representing the logical qubits and the dots connecting the two representing a $\hat{U}^{CZ}$.

The $\hat{U}^{CZ}$ operations arise naturally from the choice of logical encoding in the graph. Measuring qubits zero and one in the basis $M(\alpha)$ and $M(\beta)$ respectively will, without Pauli by-products, propagate $\hat{H}\hat{R}_{z_1}(\alpha)$ and $\hat{H}\hat{R}_{z_2}(\beta)$. Then, measuring qubits two and three in the basis $M(\pi)$ is equivalent to applying the last $\hat{H}\hat{Z}$ operations of the inversion around the mean and doing the final measurement, effectively finishing the algorithm. Further, the tagged item is given by $\left|a_2\right\rangle_{L_1}\left|a_3\right\rangle_{L_2}$, with $a_2/a_3 = 0$ for measurement outcome $M_+(\pi)$ on qubit two and three, and $a_2/a_3 = 1$ for measurement outcome $M_-(\pi)$ on qubit two and three. The Pauli by-products from the first two measurements are accounted for in final measurement outcomes, and are $\hat{Z}^{a_0}\hat{X}^{a_1}$ on logical qubit one and $\hat{Z}^{a_1}\hat{X}^{a_0}$ on logical qubit two [45]. These will simply switch the tagged item to $|a_2 \oplus a_1\rangle_{L_1} |a_3 \oplus a_0\rangle_{L_2}$.

To better see how the algorithm works, we derive the algorithm for the case of no Pauli by-products and tagging the $|1\rangle_{L_1} |1\rangle_{L_2}$ item. First, the box graph is written as:

$$\left|\psi^{Box}_{N=4}\right\rangle = \frac{1}{2}(|0\rangle_0 |+\rangle_1 |0\rangle_2 |+\rangle_3 + |0\rangle_0 |-\rangle_1 |1\rangle_2 |-\rangle_3 + |1\rangle_0 |-\rangle_1 |0\rangle_2 |-\rangle_3 + |1\rangle_0 |+\rangle_1 |1\rangle_2 |+\rangle_3). \tag{4.11}$$

The first two measurements of the algorithm corresponds to applying the following projection operator to $\left|\psi^{Box}_{N=4}\right\rangle$:

$$\hat{M}_{0,+}(\alpha) \otimes \hat{M}_{1,+}(\beta) \otimes \hat{\mathbb{1}}_2 \otimes \hat{\mathbb{1}}_3 = (|+,\alpha\rangle_0 \langle +,\alpha|_0) \otimes (|+,\beta\rangle_1 \langle +,\beta|_1) \otimes \hat{\mathbb{1}}_2 \otimes \hat{\mathbb{1}}_3. \tag{4.12}$$

Then, the final two measurements on qubit two and three are performed in the following basis:

$$|\psi\rangle_{00} = |-\rangle_2 |-\rangle_3 \,, \tag{4.13}$$

$$|\psi\rangle_{01} = |-\rangle_2 |+\rangle_3 \,, \tag{4.14}$$

$$|\psi\rangle_{10} = |+\rangle_2 |-\rangle_3 \,, \tag{4.15}$$

$$|\psi\rangle_{11} = |+\rangle_2 |+\rangle_3 \,. \tag{4.16}$$

Choosing to tag $|1\rangle_{L_1} |1\rangle_{L_2}$ sets $\alpha = \beta = 0$ and the first measurements projects the box state to:

$$|\psi\rangle_T = \frac{\hat{M}_0(0) \otimes \hat{M}_1(0) \otimes \hat{\mathbb{1}}_2 \otimes \hat{\mathbb{1}}_3 \left|\psi^{Box}_{N=4}\right\rangle}{\sqrt{\langle\psi^{Box}_{N=4}|\, \hat{M}_0(0) \otimes \hat{M}_1(0) \otimes \hat{\mathbb{1}}_2 \otimes \hat{\mathbb{1}}_3 \,|\psi^{Box}_{N=4}\rangle}} = \tag{4.17}$$

$$\frac{|+\rangle_0 |+\rangle_1 |0\rangle_2 |+\rangle_3 + |+\rangle_0 |+\rangle_1 |1\rangle_2 |+\rangle_3}{\sqrt{2}}, \tag{4.18}$$

subsequently the final measurements yield

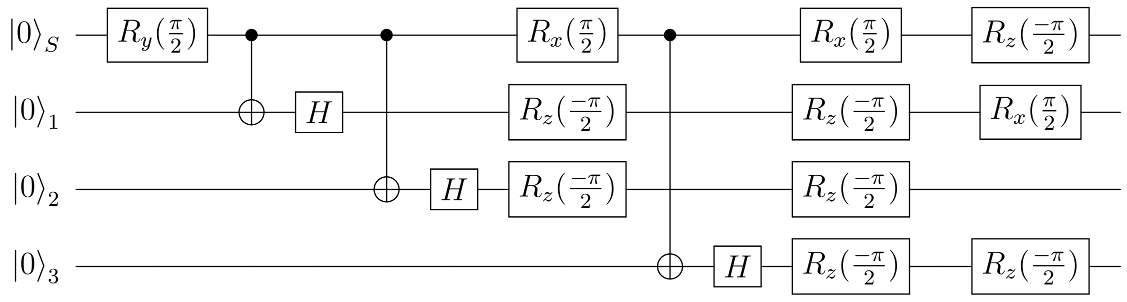
$$|\langle\psi_{11}|\psi\rangle_T|^2 = 1, \tag{4.19}$$

$$|\langle\psi_{10}|\psi\rangle_T|^2 = |\langle\psi_{01}|\psi\rangle_T|^2 = |\langle\psi_{00}|\psi\rangle_T|^2 = 0, \tag{4.20}$$

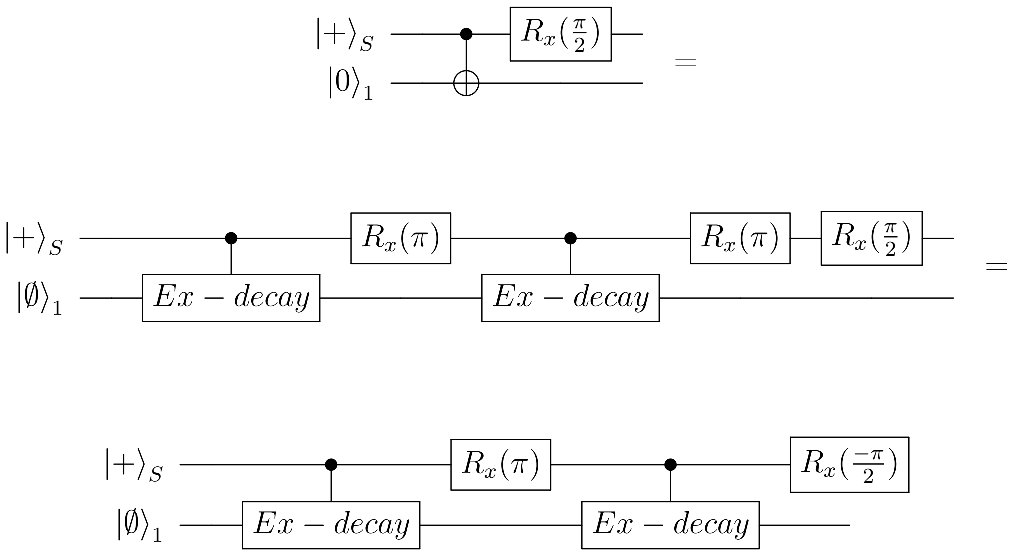
giving the probability of measuring the tagged item $P_{11} = 1$.

4.1.3 Simulating Grover's algorithm

The square graph is indeed accessible with our QD and except for generating photons, which amounts to performing a CNOT between spin and photon, three local implementations are needed. Below is the circuit to generate the square graph



The $\hat{R}_x(\frac{\pi}{2})$ and $\hat{R}_z(\frac{-\pi}{2})$ comes from applying local complementation (LC). Furthermore, all the single-qubit operations on the photons are accounted for in the final measurement basis. Before moving on to the simulation of the algorithm, there is one thing to be noted. The CNOT's encodes controlled excitation's and decays, and $R_x(\pi)$ on the spin. Whenever LC is applied on the spin, the last $R_x(\pi)$ in the CNOT generation can be concatenated with $\hat{R}_x(\frac{\pi}{2})$ to a total of $\hat{R}_x(\frac{-\pi}{2})$, which is illustrated in the circuits below. This reduces the total rotation pulse duration, and thus also the errors that it induces.



We end this subsection with simulating Grover's algorithm as described above with the error model introduce in section 3.3. The graph state density matrix is sampled with 1000 trajectories. We simulate for three different parameter settings: (1) All errors turned of except for $T_2^* = 23.2$ ns, (2) experimental parameters following table 3.1 with $\theta = 0.1°$, (3) the same setting as in (2), except for the spin flip rate changed to $\kappa = 0.21$ ns^{-1}. These three different settings represents an ideal setting, a realistic setting and a worst setting.

Histograms with probabilities for all three cases without Pauli by-products are shown in figures 4.4, 4.5, 4.6, respectively. Given the large error induced in an experimental setting, the algorithm is fairly robust to errors with an average success probability above 75% seen in figure 4.5. Furthermore, in an ideal setting the performance reaches an average above 95%, and, as expected, by increasing the spin flip rate by an order of magnitude the performance reduces drastically. In the worst case, the tagging process is diffused and the performance is not better than simply guessing.

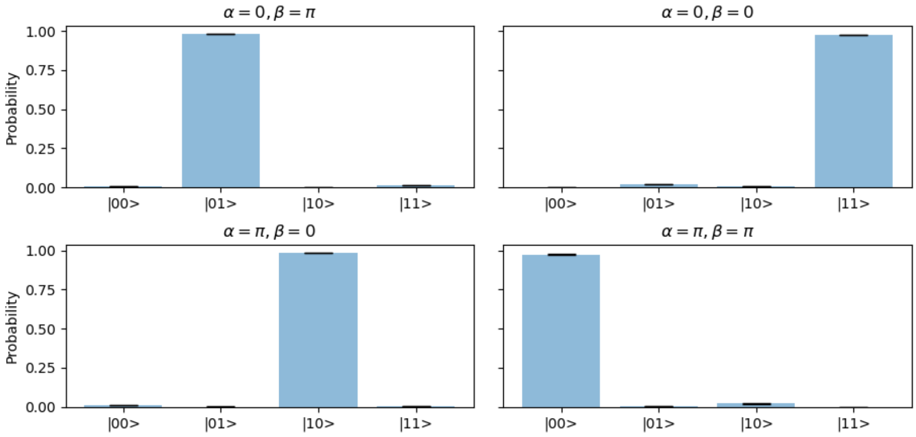


Figure 4.4: Running Grover's algorithm with all experimental parameters set to zero except $T_2^* = 23.2$ ns. The error bars indicate the uncertainties from the 1000 Monte-Carlo trajectories. The title of each subplot indicate the item we aimed to tag.

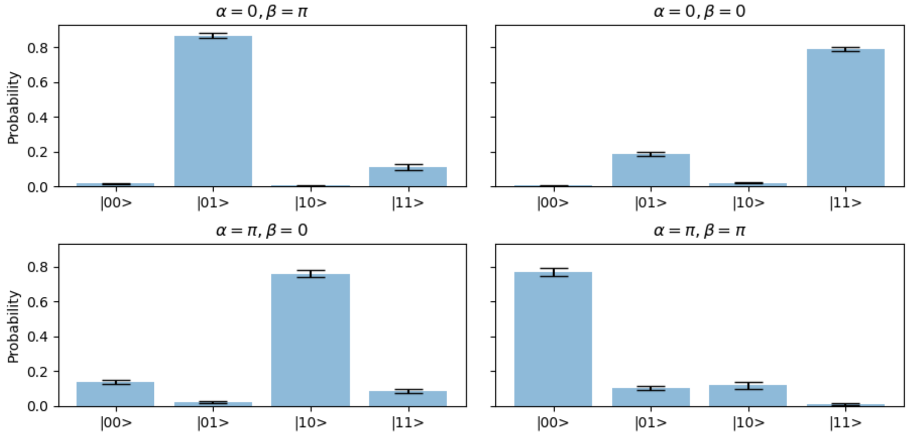


Figure 4.5: Running Grover's algorithm with experimental parameters following table 3.1, and $\theta = 0.1°$. The error bars indicate the uncertainties from the 1000 Monte-Carlo trajectories.

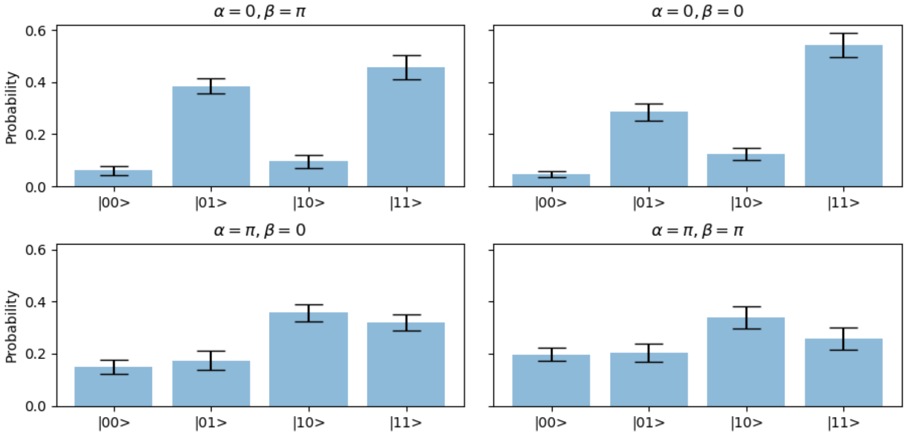


Figure 4.6: Running Grover's algorithm with experimental parameters following table 3.1 except $\kappa = 0.21$ ns^{-1}. The error bars indicate the uncertainties from the 1000 Monte-Carlo trajectories.

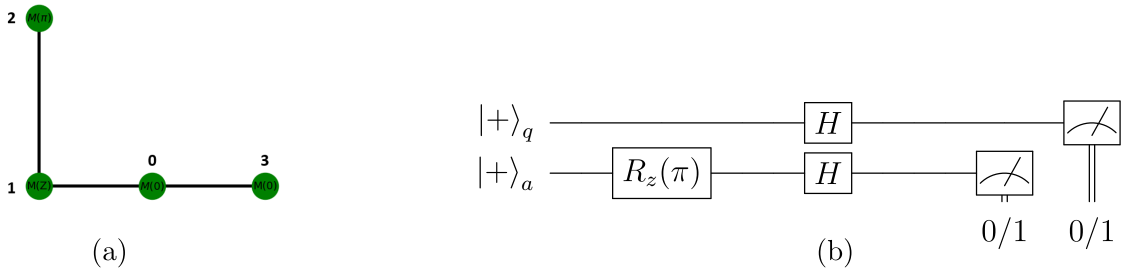
4.1.4 Measurement based Deutsch's algorithm

The 2D cluster state shown in figure 4.8a will act as our resource state to perform Deutsch's algorithm with two input qubits. This cluster state is the four qubit line graph, but twisted at one endpoint. Qubit one and two will encode logical qubit one L_1, and qubit zero and three will encode logical qubit two L_2. Logical qubit one will act as our query qubit (q), while logical qubit two will act as our ancilla qubit (a). The scheme of the algorithm follow that of Ref. [45] and is shown in figures 4.8b and 4.7b. The outline, in the case of a constant function, is as follows:

(i) The query and ancilla qubits are prepared in the state $|+\rangle_q |-_a\rangle$, which is done by applying $\hat{R}_z(\pi)$ to the ancilla qubit.

(ii) The black box does nothing.

(iii) Hadamard gates are applied to both query and ancilla qubits and measurements are performed in the computational basis.



(a)

(b)

Figure 4.7: (a) The line graph with labels corresponding to physical qubits and the corresponding measurement basis for performing Deutsch's algorithm with a constant function. (b) The quantum circuit version of Deutsch's algorithm with a constant function. Here q and a represents the query and ancilla qubit, respectively.

By measuring qubit two in the basis $M(\pi)$, with a plus state outcome, will propagate $\hat{H}\hat{R}_z(\pi)$ onto our ancilla qubit. Then, measuring qubit zero in the basis $M(Z)$ removes the $\hat{U}^{CZ}$ between the ancilla and query qubit. Finally, to compensate for the last Hadamard gate on the query qubit, the final measurement is performed in the bases $M(0)$ and $M(Z)$ for qubit three and one, respectively. Accounting for the Pauli by-products is done by noting that the final output state of the algorithm is $|a_3 \oplus a_0\rangle\, |a_1 \oplus a_2\rangle$, with the $a_i's$ for $i = \{0, 1, 2, 3\}$ representing the measurement outcomes of the individual qubits.

Implementing Deutsch's algorithm for a balanced function, the outline is changed to the following:

(i) The query and ancilla qubit are prepared in the state $|+\rangle_q\, |-\rangle_a$, which is done by applying $\hat{R}_z(\pi)$ to the ancilla qubit.

(ii) The black box performs a $\hat{U}^{CNOT}$ between query and ancilla qubit.

(iii) Hadamard gates are applied to both the query and the ancilla qubit and measurements are performed in the computational basis.

Preparing the state $|+\rangle_q\, |-\rangle_a$ is again done by measuring qubit two in the basis $M(\pi)$ propagating $\hat{H}\hat{R}_z(\pi)$ to the ancilla qubit. Then, measuring qubit zero in the basis $M(\frac{\pi}{2})$ will mediate $\hat{R}_{z_q}(\frac{-\pi}{2})\hat{R}_{z_a}(\frac{-\pi}{2})\hat{U}^{CZ}_{qa}$, and using the commutation relation $\hat{U}^{CZ}_{qa}\hat{H}_a = \hat{H}_a\hat{U}^{CNOT}_{qa}$, the full sequence of gates from the first two measurements amounts to $\hat{R}_{z_q}(\frac{-\pi}{2})\hat{R}_{z_a}(\frac{-\pi}{2})\hat{U}^{CNOT}_{qa} \otimes \hat{R}_z(\pi)_a \mathbb{1}_q$. To compensate for $\hat{R}_z(\frac{-\pi}{2})$ on the ancilla qubit and $\hat{H}\hat{R}_z(\frac{-\pi}{2})$ on the query qubit, measurements on qubit one and three are done in bases $M(\frac{-\pi}{2})$ and $M(Z)$, respectively. The Pauli by-products are again accounted for in the final measurement by observing the output state of the algorithm is $|a_3 \oplus a_0 \oplus a_2\rangle\, |a_1\rangle$ [45].

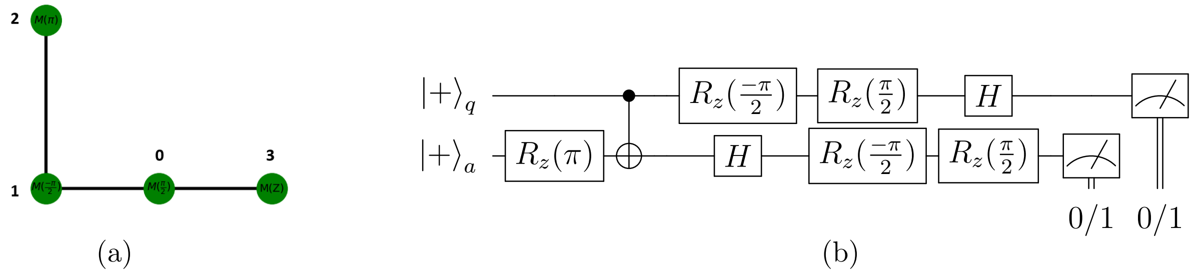


Figure 4.8: (a) The line graph with labels corresponding to physical qubits and the corresponding measurement basis for performing Deutsch's algorithm with a balanced function. (b) The quantum circuit version of Deutsch's algorithm with a balanced function. Here, q and a represents the query and ancilla qubit, respectively.

As an instructive example, we work out the measurement sequence for a constant function without Pauli by-products. First, the line graph is written as

$$|\psi^{Line}_{N=4}\rangle = \frac{1}{2}(|+\rangle_0 |0\rangle_1 |0\rangle_2 |+\rangle_3 + |+\rangle_0 |0\rangle_1 |1\rangle_2 |-\rangle_3 + |-\rangle_0 |1\rangle_1 |0\rangle_2 |+\rangle_3 + |-\rangle_0 |1\rangle_1 |1\rangle_2 |-\rangle_3). \tag{4.21}$$

Next, measuring qubit two and qubit zero amounts to applying the following projection operator:

$$\hat{M}_{0,+}(Z) \otimes \hat{\mathbb{1}}_1 \otimes \hat{M}_{2,+}(\pi) \otimes \hat{\mathbb{1}}_3 = (|0\rangle_0 \langle 0|_0) \otimes \hat{\mathbb{1}}_1 \otimes (|-\rangle_2 \langle -|_2) \otimes \hat{\mathbb{1}}_3, \tag{4.22}$$

which projects the line graph to

$$\begin{aligned} |\psi\rangle_D = \hat{M}_0(Z) \otimes \hat{\mathbb{1}}_1 \otimes \hat{M}_2(\pi) \otimes \hat{\mathbb{1}}_3 \left|\psi^{Line}_{N=4}\right\rangle = \\ \frac{1}{2} |0\rangle_0 |0\rangle_1 |-\rangle_2 |+\rangle_3 - |0\rangle_0 |0\rangle_1 |-\rangle_2 |-\rangle_3 + |0\rangle_0 |1\rangle_1 |-\rangle_2 |+\rangle_3 - |0\rangle_0 |1\rangle_1 |-\rangle_2 |-\rangle_3 \,. \end{aligned} \tag{4.23}$$

Finally, measuring qubits three and one in the following basis:

$$|\psi\rangle_{00} = |+\rangle_1 |0\rangle_3 \,, \tag{4.24}$$

$$|\psi\rangle_{01} = |+\rangle_1 |1\rangle_3 \,, \tag{4.25}$$

$$|\psi\rangle_{10} = |-\rangle_1 |0\rangle_3 \,, \tag{4.26}$$

$$|\psi\rangle_{11} = |-\rangle_1 |1\rangle_3 \,, \tag{4.27}$$

yields:

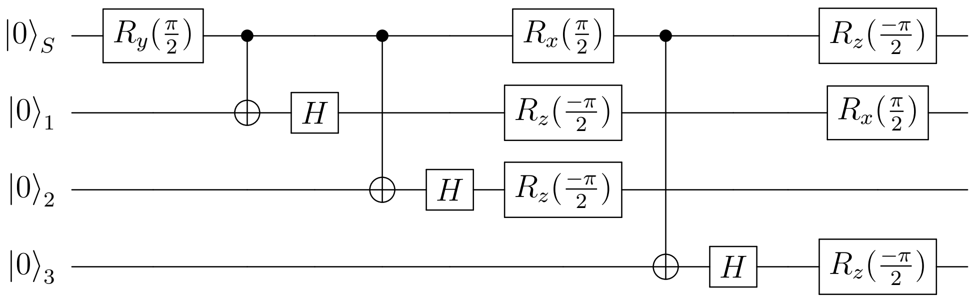
$$|\langle \psi_{01} | \psi \rangle_D|^2 = 1, \tag{4.28}$$

$$|\langle \psi_{00} | \psi \rangle_D|^2 = |\langle \psi_{10} | \psi \rangle_D|^2 = |\langle \psi_{11} | \psi \rangle_D|^2 = 0, \tag{4.29}$$

giving unit probability of measuring the correct output state[4].

4.1.5 Simulating Deutsch's algorithm

The line graph needed for Deutsch's algorithm can be generated with our QD. Compared to the square graph, only two LC-operations are needed and with one of them on the spin, which results in overall more spin rotations than the square graph. The circuit below generates the line graph.



For the simulations we consider the same three settings as for Grover's, and sample the graph state density matrix with 1000 trajectories. Histograms with probabilities of each measurement outcome for the case of no Pauli by-products are shown in figures 4.9, 4.10, 4.11 for all three settings for both a balanced and a constant function. The success probability vary slightly between balanced and constant function, and on average the success probability is lower than for Grover's algorithm. To an extent, this can be attributed to the lower fidelity of the generated line graph compared to the square graph, which is due

[4] Remember, for a constant function the output state should be $|0\rangle_a |1\rangle_q$.

to more spin rotations during its generation.

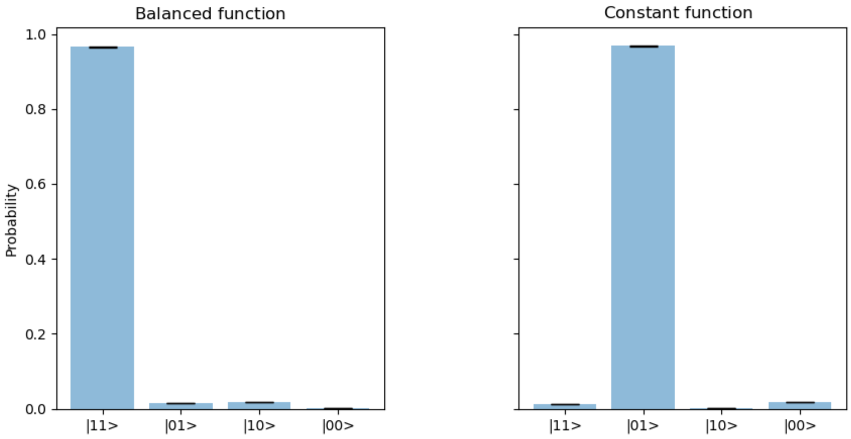


Figure 4.9: The figures show the probability of measuring the correct outcome for a constant (left) and a balanced (right) function. This is with all experimental parameters set to zero except $T_2^* = 23.2$ ns. The error bars indicate the uncertainties from the 1000 Monte-Carlo trajectories.

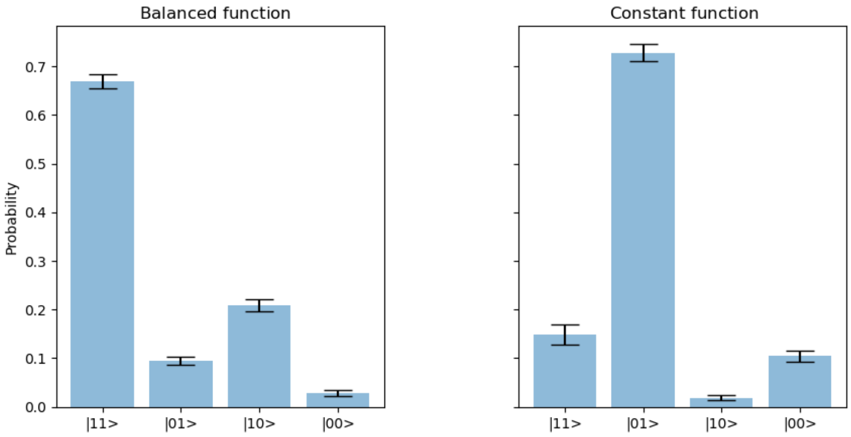


Figure 4.10: Figures show the probability of measuring the correct outcome for constant (left) and balanced (right) function with experimental parameters following table 3.1. The error bars indicate the uncertainties from the 1000 Monte-Carlo trajectories

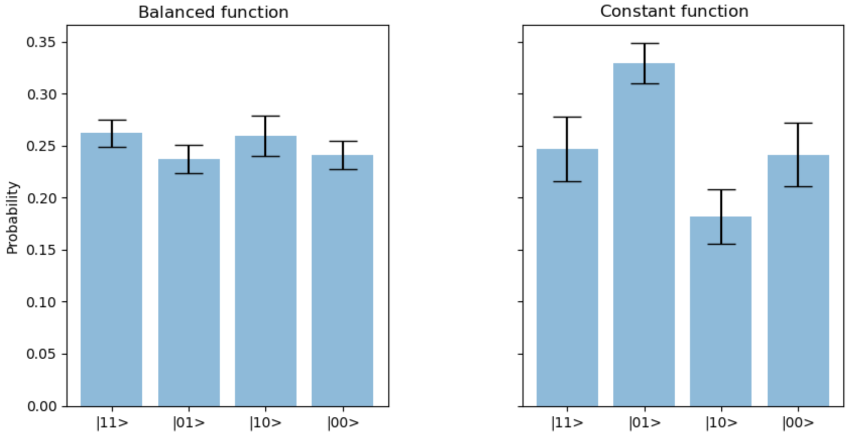


Figure 4.11: Figures show the probability of measuring the correct outcome for constant (left) and balanced (right) function, with experimental parameters following table 3.1 except with spin flip rate changed to $\kappa = 0.21$ ns^{-1}. The error bars indicate the uncertainties from the 1000 Monte-Carlo trajectories

4.2 Photonic variational quantum eigensolver

This section presents how one can construct a VQE algorithm using the QD graph state generator introduced in section 3.4. This is a generalisation of the protocol presented in Ref. [46], also referred to as the *photonic-variational quantum eigensolver* (P-VQE). The P-VQE is different from the general VQE in the sense that we do not have a universal set of gates, and are thus restricted to a specific set of states which we can generate. With the P-VQE, the Hamiltonians whose ground state we wish to solve for constitutes of two terms $\hat{H} = \hat{H}_a + \hat{V}$. Here $\hat{H}_a$ is an unperturbed term and $\hat{V}$ represents a perturbation. Given a graph state the unperturbed Hamiltonian is constructed by the sum of all its stabilizer generators $\mathcal{S} = \langle \hat{g}_1, .., \hat{g}_N \rangle$,

$$\hat{H}_a = -\sum_{\hat{g}_j \in \mathcal{S}} c_j \hat{g}_j, \tag{4.30}$$

and the perturbation takes the shape,

$$\hat{V} = \sum_{i}^{N} \lambda_i \hat{Z}_i, \tag{4.31}$$

where λ_i indicates the strength of the perturbation. We know the ground state of $\hat{H}_a$ is the corresponding graph state $|G\rangle$, and this will serve as our initial guess of the ansatz state. If $\hat{V}$ is small enough (i.e. $\{\lambda_i\}$ are small) it is reasonable to assume the ground state of the perturbed Hamiltonian $\hat{H}$ is in the neighbourhood of $|G\rangle$ [46]. As in Ref. [46] the optimisation parameters $\boldsymbol{\phi}$ are the angle of the initial rotation pulse and the angle of the last rotation pulse of each generated photon, which is depicted for the GHZ generation in figure 4.12[5]. This ensures polynomial scaling of $\boldsymbol{\phi}$ with the number of qubits.

[5] The figure 4.12 generalises to different graph states by incorporating $\hat{U}^{LC}$ in between generation of photons.

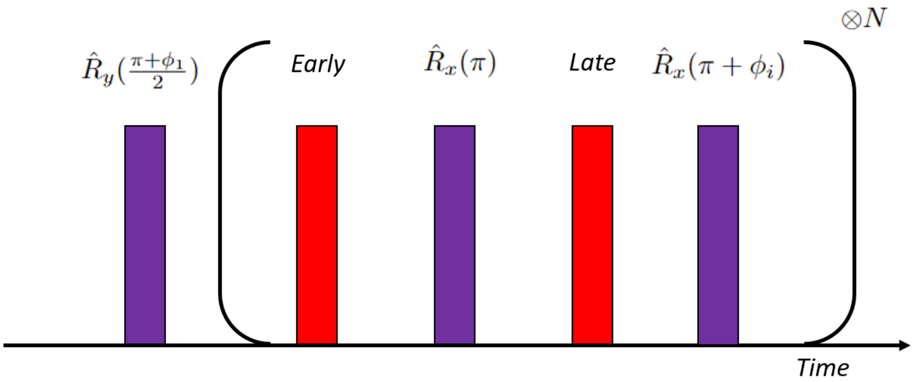


Figure 4.12: The figure illustrates the pulse sequence for generating the N-qubit GHZ state with the optimisation parameters $\boldsymbol{\phi}$.

The example we shall extensively study in this text is the two-qubit GHZ state, which is locally equivalent to the two-qubit star graph. We choose the two-qubit GHZ state because it presents the smallest Hamiltonian of our graph state VQE algorithms. The unperturbed Hamiltonian of the two-qubit GHZ state takes the form:

$$\hat{H}_a = -\hat{X}_1\hat{X}_2 - \hat{Z}_1\hat{Z}_2. \tag{4.32}$$

In the two-qubit case there are two approaches to the VQE protocol. The first approach follows the general outline above with the first rotation and last rotation of each generated photon as optimisation parameters. However, one can also remove the last rotation pulse by re-encoding the photonic qubits as $|e\rangle_p = |0\rangle_p$ and $|l\rangle_p = |1\rangle_p$. While this reduces the number of rotations and thus the errors that accompanies, it also removes one optimisation parameter. The modified pulse sequence to generate the ideal two-qubit GHZ state for the second approach follows:

$$|0\rangle_S |\emptyset\rangle_p \xrightarrow{\hat{R}_y(\frac{\pi}{2})} \frac{(|0\rangle_S + |1\rangle_S)|\emptyset\rangle_p}{\sqrt{2}} \xrightarrow{Excite-and-decay} \tag{4.33}$$

$$\frac{(|0\rangle_S |\emptyset\rangle_p + |1\rangle_S |e\rangle_p)}{\sqrt{2}} \xrightarrow{\hat{R}_x(\pi)} \frac{(|1\rangle_S |\emptyset\rangle_p + |0\rangle_S |e\rangle_p)}{\sqrt{2}} \xrightarrow{Excite-and-decay} \tag{4.34}$$

$$\frac{(|1\rangle_S |l\rangle_p + |0\rangle_S |e\rangle_p)}{\sqrt{2}} = \frac{(|0\rangle_S |0\rangle_p + |1\rangle_S |1\rangle_p)}{\sqrt{2}}. \tag{4.35}$$

Note, we have to keep the above $\hat{R}_x(\pi)$ pulse to a pi-pulse, since this act as our Echo pulse. That is, we can not use the angle of this rotation as an optimisation parameter. In the next section we shall simulate the VQE algorithm using both approaches.

4.2.1 Simulating a two qubit VQE algorithm

We choose *Nelder-Mead* [47] as our optimisation algorithm, as it is easier to implement experimentally compared to *simulated annealing* [48] or *basin-hopping* [49]. Moreover, we consider three different cases of perturbation parameter settings, which are given by: (1) $\lambda_1 = \lambda_2 = \lambda$, (2) $\lambda_1 = \lambda$ and $\lambda_2 = 0$ and (3) $\lambda_1 = -0.3\lambda$ and $\lambda_2 = 0.6\lambda$.

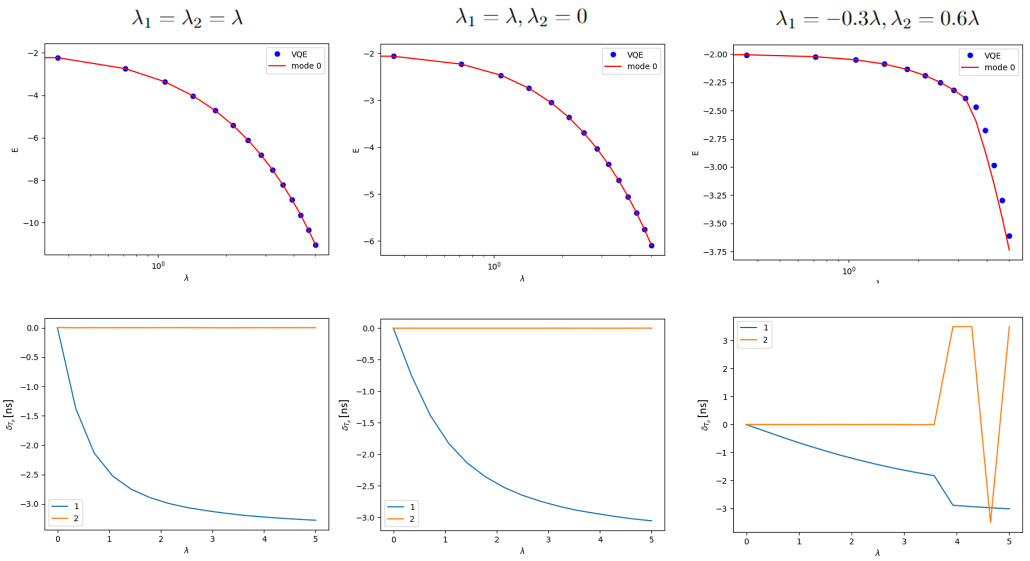


Figure 4.13: The above three figures show the results of running the VQE algorithm for all three cases without errors. The y-axis shows the energy E, the x-axis the strength of the perturbation λ and *mode 0* is referring to the ground state energy from diagonalizing the Hamiltonian. The three figures below show the evolution of the optimisation parameters as functions of the perturbation strength. Note that ϕ_1 and ϕ_2 (indicated by *1* and *2*, respectively) directly corresponds to changing the pulse duration, which is shown as δT_p on the y-axis.

We start by simulating all of the above cases in an ideal setting with no errors using the first approach with two optimisation parameters. Looking at figure 4.13 we see that in the ideal case the VQE manages to find the ground state energy. Furthermore, it seems that mainly the first variational parameter ϕ_1 is changing during the algorithm for the first two cases, while in the third case the second parameter evolves as well. Overall, the ansatz state is expressible enough to find the ground state for all three cases and we move on to include errors using the realistic error model.

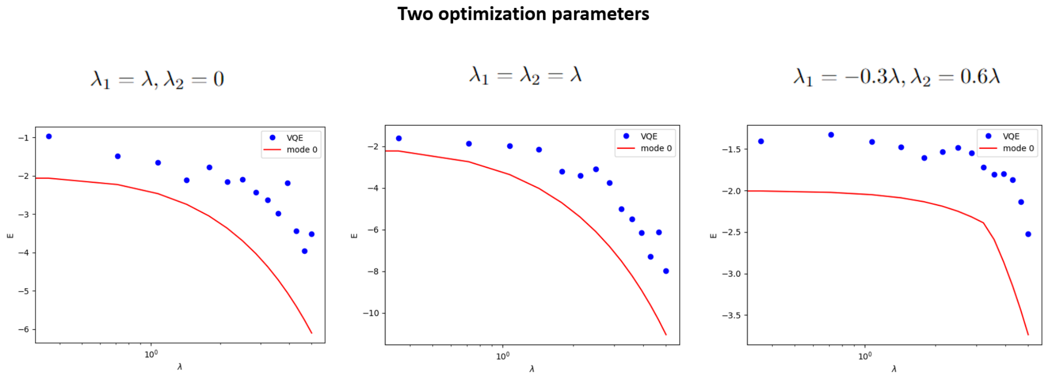


Figure 4.14: The figures show the results of running the VQE algorithm for all three cases with the error model in 3.3, using two optimisation parameters.

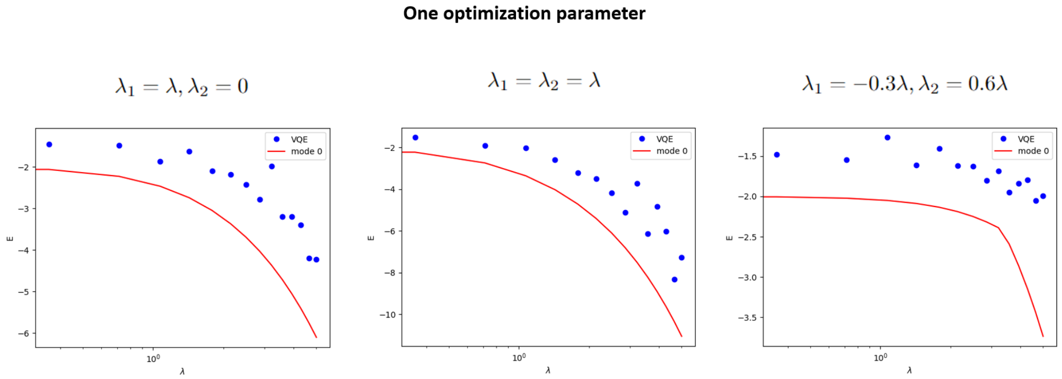


Figure 4.15: The figures show the results of running the VQE algorithm for all three cases with the error model in 3.3, using one optimisation parameter.

The figures 4.14 and 4.15 show the VQE performance for both one and two optimisation parameters with experimental parameters set to the values given in table 3.1. Looking at figures 4.14 and 4.15 it is difficult to see the difference in performance using one or two optimisation parameters. However, the error slightly decreases in the two first cases when using only one optimisation parameter instead of two, where for case (1) it goes from $37 \pm 7\% \rightarrow 33 \pm 6\%$ and for case (2) $38 \pm 8\% \rightarrow 34 \pm 6\%$. The increase in performance, although small, is expected from looking at the evolution of the optimisation parameters in figure 4.13 and accounting for the large errors induced by rotation pulses. For case (3) the error is instead larger for one optimisation parameter, with the error going from $31 \pm 6\% \rightarrow 29 \pm 6\%$ by including the second optimisation parameter. The decrease in performance indicates, in this case, that the second optimisation parameter is necessary for the expressability of the ansatz state.

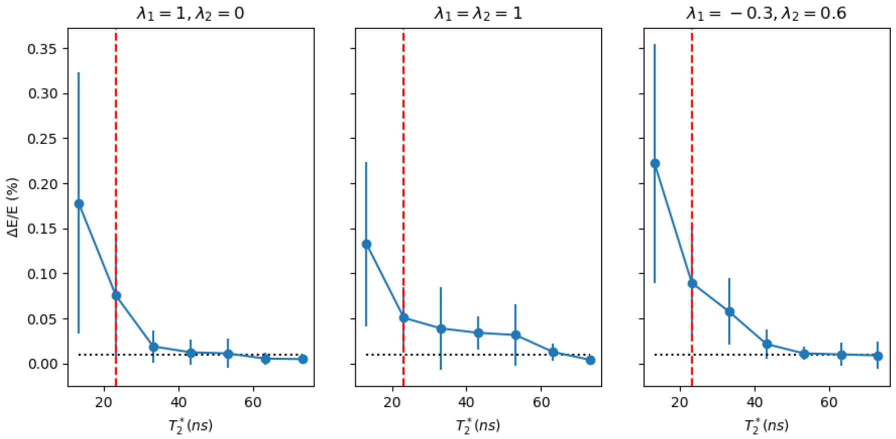


Figure 4.16: The figure illustrates three plots of the error in the VQE algorithm against T_2^* for three different perturbation settings. The error bars indicate the uncertainty from 10 different optimisation runs. The red striped line indicates the position of $T_2^* = 23.2$ ns, and the black dotted line the position of $\Delta E/E = 1\%$.

The errors using realistic experimental parameters are rather large. Although we do not have an error threshold like the chemical accuracy in quantum chemistry, we can explore what the constraints are on the experimental parameters to achieve an error smaller than $\Delta E/E < 1\%$. In GaAs/InAs quantum dots the T_2^* parameter will always be present due to the nuclear spin of the constitutes[6]. Hence, we choose to turn off all other experimental parameters, and scan $\Delta E/E$ over different values of T_2^*. This is illustrated in figure 4.16 for all three cases with $\lambda = 1$. Figure 4.16 imply a $T_2^* = 73.2$ ns would be needed to be able to achieve $\Delta E/E < 1\%$ for all three different perturbations.

Summarising, we have investigated the smallest instance of our graph state VQE algorithm. Sacrificing one optimisation parameter for overall less rotations resulted in increased performance for two out of three Hamiltonians. Overall, with realistic experimental parameters the performance in all three cases is rather poor, where in the best case an average error $\sim 29 \pm 6\%$ is achieved. As we do not have a chemical accuracy to relate to we set the error threshold to 1% and illustrated $T_2^* = 73.2$ ns plus all other errors turned off would be needed to reach the threshold for all three versions of the perturbation.

[6] That is why it was never turned off in any of the three setting when simulating Grover's and Deutsch's algorithm

Chapter 5

Error and loss-tolerant applications with quantum dot generated graph states

This chapter presents a study of two applications using graph codes. The graph codes are built to withstand gate errors and qubit loss. Qubit loss is a peculiar type of error, very common in quantum photonics, where the qubit state is irreversibly erased with no chance of recovering it with error-correction. Still, we will see how quantum codes can be built to protect the encoded quantum state from qubit loss. The chapter starts by introducing *measurement based error-correction*, which shows how the quantum information encoded in a graph code can be recovered by only destructive qubit measurements in a way tolerant to noise, for both gate and loss errors. This is then applied to a scheme for indirectly reading out a spin qubit state and a scheme for quantum key distribution.

5.1 Error-corrected and loss-tolerant logical measurements

For near term applications, there are several limitations to implement ancilla qubits for measuring stabilizers, as presented in section 1.3.5. The two main limitations are: (1) resources are scare and (2) two-qubit operations between photons are probabilistic. A more friendly approach for quantum photonics consists in the case of where only single-qubit measurements on the code qubits, readily and deterministically available with photons, are used to recover or process the encoded information. For simplicity, we will here focus on applications that require measuring a logical operator of a graph code. In the next subsection we will describe how to build graph codes and decode them in this scenario.

5.1.1 Building a code for destructive measurements

Constructing a graph code for destructive measurement based error-correction follows the principles introduced in section 2.2.2, but with one addition, a measurement pattern. As discussed in section 2.2.2 the subset I of the graph state $|G\rangle$ defines the logical operators and the stabilizer generators of the code. However, as measurements are destructive we can only infer information of a subset of the stabilizers and the logical operator of the graph code simultaneously. This is summed up below:

Stabilizer and logical operator matching:

Consider a graph state $|G\rangle$ constructed from qubits $\mathcal{O}$, which along with I defines a graph code $\mathcal{T}$ with stabilizers $\mathcal{S}_{\mathcal{C}}$. If there is a subset of stabilizers $\mathcal{S}_{\mathcal{R}} \in \mathcal{S}_C$ and a logical operator L that satisfy:

1. *All stabilizers commute with the logical operator at each output qubit: $[\hat{S}_i^{[j]}, \hat{L}^{[j]}] = 0$ $\forall \hat{S}_i \in \mathcal{S}_{\mathcal{R}}$ and $\forall j \in \mathcal{O}$*
2. *All stabilizers commute on all qubits $\mathcal{O}$: $[\hat{S}_i^{[j]}, \hat{S}_k^{[j]}] = 0$ $\forall \hat{S}_i, \hat{S}_k \in \mathcal{S}_{\mathcal{R}}$ and $\forall j \in \mathcal{O}$,*

then the value of all the stabilizers in $\mathcal{S}_{\mathcal{R}}$ and the logical operator L of $\mathcal{T}$ can be inferred simultaneously from a measurement pattern M of collapsing single-qubit measurements on the qubits $\mathcal{O}$. The stabilizers in $\mathcal{S}_{\mathcal{R}}$ still forms a stabilizer group and error-correction can proceed as usual.

This follows directly from the transformation of stabilizers under measurements of single Pauli operators discussed in Appendix B. Thus, apart from choosing a graph state $|G\rangle$ and the subset I, one has to define a measurement pattern to complete the graph code. The measurement pattern plays a crucial role in the error-correcting capabilities of the code, and the procedure for choosing the best measurement pattern is done in three steps and outlined below:

1. Given a graph code $\mathcal{T}$ and a logical operator L, find all measurement patterns compatible with L. That is, find all measurement patterns which allows measuring L, i.e. all qubits i where $L^{[i]}$ acts with a Pauli operator different from identity, $M^{[i]} = L^{[i]}$.
2. Select the measurement patterns with the most available stabilizers, i.e. $|\mathcal{S}_R|$ is largest.
3. From these measurement patterns, choose the pattern with the logical operator of lowest weight. With weight referring to the number of Pauli operators different from identity.

Each measured stabilizer gives us one bit of information. Hence, the idea behind the best measurement pattern is: The most amount of measured stabilizers will give the most amount of information and the lesser the weight of the logical operator the lower the probability for an error.

5.1.2 Decoder

As syndromes and errors do not map one-to-one, one has to construct a decoder whose objective is to decide where and what error (errors) is (are) most likely given the syndrome.

There are many examples of different decoders for various codes [50, 51, 52, 53], and we here aim to construct a decoder suitable for logical measurements based on destructive measurements of the physical qubits. Because we restrict ourselves to codes with a small number of qubits, we do not care about the computational efficiency of the decoder, but only care about maximising the noise tolerance of the decoder.

The probability of an error in the measurement of the logical operator L can be written as:

$$p(\epsilon_L) = \sum_N p(\epsilon_L|S(N))p(S(N)), \tag{5.1}$$

where $S(N) = (V_1, .., V_N)$ are the measured values of all stabilizer generators in $\mathcal{S}_R$ (i.e. the syndrome), with V_i being the measured value of stabilizer generator $\hat{g}_i$. Furthermore, ϵ_L stands for the logical error, which is different from the physical error. From here, the error correction is performed using maximal likelihood estimation on the conditional probability of a logical error $p(\epsilon_L|S(N))$,

$$p(\epsilon_L|S(N)) \longrightarrow p_{\text{corrected}}(\epsilon_L|S(N)) = \min(p(\epsilon_L|S(N)), 1 - p(\epsilon_L|S(N))). \tag{5.2}$$

This results in the final probability of an error in the logical measurement:

$$p(\epsilon_L) = \sum_N p_{\text{corrected}}(\epsilon_L|S(N))p(S(N)). \tag{5.3}$$

This way of error-correcting the logical measurement can be seen as flipping the measurement outcome of L if the probability of an error given the syndrome $p(\epsilon_L|S(N))$ is large (above 50%).

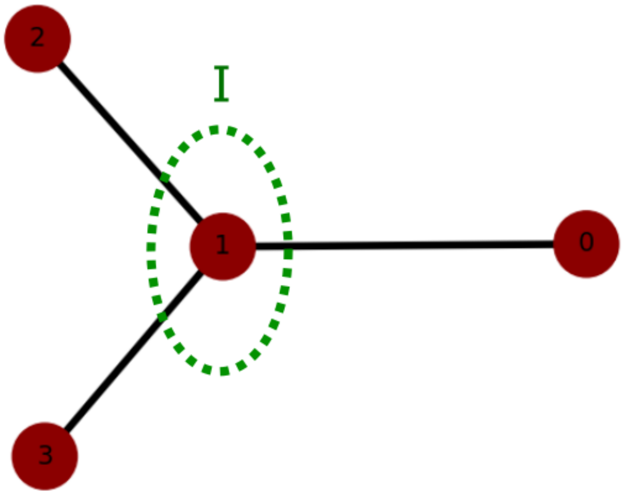
5.1.3 Adding photon loss

In the context of loss-tolerant logical measurements we are only concerned with being able to measure a logical operator while the graph code is subjected to unheralded photon loss[1]. As the measurement pattern is carried out and photon loss is detected, a new measurement pattern has to be found for the logical operator with $\hat{\mathbb{1}}$ on the lost qubit and same Pauli operators on the already measured qubits. When no new measurement pattern can be found, the measurement fails and the logical qubit is considered lost. However, for a graph code with N stabilizer generators there exists $\mathcal{O}(2^N)$ different possible logical operators for a given basis[2], and thus there is a multitude of measurement patterns to be chosen from. Ultimately, logical operators with a small weight $|L|$ requires less qubits to be measured, and thus a code with many such logical operators will be loss-tolerant.

We consider an example to clarify the measurement procedure and the loss-tolerant properties of graph codes. Consider the four qubit star graph code shown in figure 5.1. In this code, there are four equally loss-tolerant measurement patterns for logical-X: $\{M_1 = \hat{\mathbb{1}}_0\hat{Z}_1\hat{\mathbb{1}}_2\hat{\mathbb{1}}_3, M_2 = \hat{\mathbb{1}}_0\hat{\mathbb{1}}_1\hat{X}_2\hat{\mathbb{1}}_3, M_3 = \hat{\mathbb{1}}_1\hat{\mathbb{1}}_2\hat{X}_3\hat{\mathbb{1}}_4, M_4 = \hat{X}_0\hat{\mathbb{1}}_1\hat{\mathbb{1}}_2\hat{\mathbb{1}}_3\}$. If M_1 is initiated and photon one is detected lost, one can easily change pattern to M_2, M_3 or M_4. That is,

[1] Unheralded means we do not know which photon is lost until we try to measure it.

[2] Remember, a logical operator times a stabilizer is still a logical operator.



$$\begin{aligned}\overline{X} &= \hat{\mathbb{1}}_0\hat{Z}_1\hat{\mathbb{1}}_2\hat{\mathbb{1}}_3\\ \overline{Z} &= \hat{Z}_0\hat{X}_1\hat{Z}_2\hat{Z}_3\\ \mathcal{S} = \{\hat{g}_1 &= \hat{X}_0\hat{Z}_1\hat{\mathbb{1}}_2\hat{\mathbb{1}}_3,\\ \hat{g}_2 &= \hat{\mathbb{1}}_0\hat{Z}_1\hat{X}_2\hat{\mathbb{1}}_3,\\ \hat{g}_3 &= \hat{\mathbb{1}}_0\hat{Z}_1\hat{\mathbb{1}}\hat{X}_3\}\end{aligned}$$

Figure 5.1: The figure illustrates the graph object of a four qubit star code with the first qubit composing the subset I. Furthermore, the logical operators $\overline{X}$ and $\overline{Z}$ plus the stabilizer generators $\mathcal{S}$ are shown.

to be able to measure logical-X of the star qubit graph code only one out of four qubits needs to be measured. This yields the logical transmission $T^{\overline{X}} = 1 - p^4$, with p as the probability of photon loss. While logical-X is loss-tolerant, logical-Z is not. This can be seen from the fact that any version of a logical-Z includes Pauli operators different from identity on all qubits of the graph code. Thus, all qubits have to be measured yielding $T^{\overline{Z}} = (1-p)^4$, which makes the code very susceptible to photon loss.

Having discussed how logical measurements can be performed in an error-corrected and loss-tolerant manner, we can merge the two. Finding the optimal measurement strategy for both loss-tolerance and error-correction is compatible with each other, as the weight of the logical operator is important in both cases. To complete the merger the measurement pattern is initialised by first measuring the qubits corresponding to the logical operator and then continues with measuring the qubits necessary to infer the values of $\mathcal{S}_R$. The algorithm that decodes the graph codes for losses and gate errors simultaneously is outlined in pseudo code in Appendix C.

5.2 Indirect measurement of a spin state

The first application we will consider is indirectly measuring the spin state of a spin-photon interface system, such as the QD. Imagine the spin in a generic quantum state, $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ and we want to measure it. A possibility is to entangle the spin with a photonic qubit through a CNOT, generating the state: $|\Psi\rangle = a\,|00\rangle + b\,|11\rangle$. Measuring the photon in the Z-basis, due to perfect correlations in $|\Psi\rangle$, is equivalent to measuring the spin in the Z-basis. However, as discussed in chapter 3.3 the entanglement generation is not perfect, there are many sources of error. Hence a question arise: Can we generate error robust graph codes that can be entangled with the spin, which yield a better indirect measurement of the quantum dot than the Bell-like state $|\Psi\rangle$?

To clarify, consider if the photon in $|\Psi\rangle$ is subjected to a $\hat{X}$ error. This transforms $|\Psi\rangle \xrightarrow{\hat{X}} a\,|0\rangle_S\,|1_p\rangle + b\,|1\rangle_S\,|0\rangle_p$, and the measurement of the photon will be anti-correlated with the spin resulting in an erroneous indirect spin readout. However, imagine the spin is instead entangled with a simple repetition code $|\Psi\rangle = a\,|0\rangle_S\,|\overline{0}\rangle + b\,|1\rangle_S\,|\overline{1}\rangle = a\,|0\rangle_S\,|000\rangle + b\,|1\rangle_S\,|111\rangle$. For this code logical-Z is given by $\overline{Z} = \hat{Z}_1\hat{\mathbb{1}}_2\hat{\mathbb{1}}_3$, which is compatible with three stabilizers $\{\hat{S}_1 = \hat{Z}_1\hat{\mathbb{1}}_2\hat{Z}_3, \hat{S}_2 = \hat{Z}_1\hat{Z}_2\hat{\mathbb{1}}_3, \hat{S}_3 = \hat{\mathbb{1}}_1\hat{Z}_2\hat{Z}_3\}$. If the first photon is subjected to a $\hat{X}$ error, then: $|\Psi\rangle \xrightarrow{\hat{X}} a\,|0\rangle_S\,|100\rangle + b\,|1\rangle_S\,|011\rangle$. However, from measuring the qubits of the code we obtain the syndrome $(-1,-1,1)$ from which we detect the error on the photon and correct the graph readout. Subsequently this leads to a correct indirect measurement of the spin. Moving on, we present a procedure to entangle the spin and a graph code using the system presented in chapter 3.

5.2.1 Entangling the spin with graph codes

We consider a scheme where generating the entanglement between the spin and the graph code is done simultaneously as generating the graph state of the graph code. This is done with the same toolbox as in section 3.4 (i.e. time-bin entanglement and local complementation), but now with the spin initially in $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. As we shall demonstrate below, this restricts us to graph codes defined by graph states of subgraphs from the LC-equivalence class of the star graph.

Starting with the spin in a generic state $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ and generating N-photons as described in section 3.2 and subsequently applying $\hat{H}$ to all of them, generates the state

$$\begin{aligned}
|\Psi\rangle &= \prod_{j\in I} \hat{U}^{CZ}_{Sj}(a\,|0\rangle_S + b\,|1\rangle_S)\,|G\rangle_N \\
&= a\,|0\rangle_S\,|G\rangle_N + b\,|1\rangle_S \prod_{j\in I} Z_j\,|G\rangle_N \\
&= a\,|0\rangle_S\,|\overline{0}\rangle_N + b\,|1\rangle_S\,|\overline{1}\rangle_N .
\end{aligned} \tag{5.4}$$

Here we have chosen to represent the N-photons as a graph state $|G\rangle_N = \prod_N |+\rangle^{\otimes N}$ and the subset I as all the photons. Furthermore, $\hat{U}^{CZ}_{Sj}$ is a control phase between the spin S and photon j. From (5.4) we see that generating photons as in section 3.2 with the spin initially in $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ translates to entangling the spin with the graph code defined by the photons in the graph state of the fully disconnected graph. As in section 3.4, from here it is possible to apply $\hat{U}^{LC}$ on (5.4) to generate graph codes defined by different graph states. The transformation $|\Psi\rangle \to \hat{U}^{LC}\,|\Psi\rangle$ follows as:

Code transformation under local complementation

Consider the entangled state $|\Psi\rangle = a\,|0\rangle_S \left|\bar{0}\right\rangle_N + b\,|1\rangle_S \left|\bar{1}\right\rangle_N$*, where* $\left|\bar{0}\right\rangle_N$ *and* $\left|\bar{1}\right\rangle_N$ *are the codewords associated with the graph code defined by the graph state* $|G\rangle_N$*. With:*

1. $|G\rangle_N$ *as the graph state associated with the subgraph* $G[V_N] \subseteq G(V_{all}, E)$*, where* $G(V_{all}, E)$ *is the graph state of all particles including the spin,*
2. $|G\rangle'_N$ *as the graph state associated with the subgraph* $G'[V_N] \subseteq G'(V_{all}, E)$*, where* $G'(V_{all}, E) = U^{LC} G(V_{all}, E)$*,*
3. $\{|\psi\rangle, |\psi^{\perp}\rangle\}$ *as a generic basis and "∗" complex conjugation,*

applying $\hat{U}^{LC}$ *on* $|\Psi\rangle$ *will transform it to:* $|\Psi\rangle = a\,|\psi\rangle_S \left|\overline{\psi^*}\right\rangle'_N + b\,|\psi^{\perp}\rangle_S \left|\overline{\psi^{\perp *}}\right\rangle'_N$*, with the graph state of the graph code transformed as* $|G\rangle_N \to |G\rangle'_N$ *and the basis of the spin and graph code transformed as* $\{|0\rangle, |1\rangle\} \to \{(|\psi\rangle \,/\, |\psi^*\rangle), (|\psi^{\perp}\rangle \,/\, |\psi^{\perp *}\rangle)\}$*.*

This transformation rule is most easily shown by first rewriting (5.4) in terms of the graph state $|G\rangle_{N+1}$ of $G(V_{all}, E)$ (i.e. the full graph state of all particles $|V_{all}| = N + 1$):

$$\begin{aligned}
&\prod_{j\in I} \hat{U}^{CZ}_{Sj} \Big(\frac{(a+b)}{\sqrt{2}}\,|+\rangle_S + \frac{(a-b)}{\sqrt{2}}\,|-\rangle_S\Big)\,|G\rangle_N \\
&\quad= \frac{(a+b)}{\sqrt{2}}\,|G\rangle_{N+1} + \hat{Z}_S \frac{(a-b)}{\sqrt{2}}\,|G\rangle_{N+1}\,.
\end{aligned} \tag{5.5}$$

Applying $\hat{U}^{LC}$ on $|G\rangle_{N+1}$ follow the rules presented in section 2.1, but first has to be commuted through Z_S

$$\begin{aligned}
&\hat{U}^{LC}\Big(\frac{(a+b)}{\sqrt{2}}\,|G\rangle_{N+1} + \hat{Z}_S \frac{(a-b)}{\sqrt{2}}\,|G\rangle_{N+1)} \\
&= \frac{(a+b)}{\sqrt{2}}\,|G\rangle'_{N+1} + \prod_j \hat{P}_{S,j} \frac{(a-b)}{\sqrt{2}}\,|G\rangle'_{N+1}\,.
\end{aligned} \tag{5.6}$$

Here, $\hat{P}_{S,j} = \{\hat{X}_S, \hat{Y}_S, \hat{Z}_S\}$ and $\prod_j \hat{P}_{S,j}$ depend on the commutation between $\hat{U}^{LC}$ and $\hat{Z}_S$, which subsequently depend on the qubit U^{LC} is applied on. From here, we considered the three cases where $\prod_j \hat{P}_{S,j}$ is one of the three Pauli operators, since products of Pauli operators on the same qubit map to Pauli operators:

- **Case 1:** $\hat{P}_S = \hat{Z}_S$**.** We have

$$\begin{aligned}
|\Psi\rangle &= \frac{(a+b)}{\sqrt{2}}\,|G\rangle'_{N+1} + \hat{Z}_S \frac{(a-b)}{\sqrt{2}}\,|G\rangle'_{N+1} \\
&= \frac{(a+b)}{2}\Big(|0\rangle_S\,|G\rangle'_N + |1\rangle_S \prod_{j\in I} \hat{Z}_j\,|G\rangle'_N\Big) + \hat{Z}_S \frac{(a-b)}{2}\Big(|0\rangle_S\,|G\rangle'_N + |1\rangle_S \prod_{j\in I} \hat{Z}_j\,|G\rangle'_N\Big) \\
&= \frac{(a+b)}{2}\Big(|0\rangle_S \left|\bar{0}\right\rangle'_N + |1\rangle_S \left|\bar{1}\right\rangle'_N\Big) + \frac{(a-b)}{2}\Big(|0\rangle_S \left|\bar{0}\right\rangle'_N - |1\rangle_S \left|\bar{1}\right\rangle'_N\Big) \\
&= a\,|0\rangle_S \left|\bar{0}\right\rangle'_N + b\,|1\rangle_S \left|\bar{1}\right\rangle'_N\,.
\end{aligned} \tag{5.7}$$

- **Case 2:** $\hat{P}_S = \hat{X}_S$**.** We have

$$\begin{aligned}
|\Psi\rangle &= \frac{(a+b)}{\sqrt{2}} |G\rangle'_{N+1} + \hat{X}_S \frac{(a-b)}{\sqrt{2}} |G\rangle'_{N+1} \\
&= \frac{(a+b)}{2}(|+\rangle_S |\overline{+}\rangle'_N + |-\rangle_S |\overline{-}\rangle'_N) + \frac{(a-b)}{2}(|+\rangle_S |\overline{+}\rangle'_N - |-\rangle_S |\overline{-}\rangle'_N) \\
&= a |+\rangle_S |\overline{+}\rangle'_N + b |-\rangle_S |\overline{-}\rangle'_N .
\end{aligned} \tag{5.8}$$

- **Case 3:** $\hat{P}_S = \hat{Y}_S$**.** We have

$$\begin{aligned}
|\Psi\rangle &= \frac{(a+b)}{\sqrt{2}} |G\rangle'_{N+1} + \hat{Y}_S \frac{(a-b)}{\sqrt{2}} |G\rangle'_{N+1} \\
&= \frac{(a+b)}{2}(|y-\rangle_S |\overline{y+}\rangle'_N + |y+\rangle_S |\overline{y-}\rangle'_N) + \frac{(a-b)}{2}(|y+\rangle_S |\overline{y-}\rangle'_N - |y-\rangle_S |\overline{y+}\rangle'_N) \\
&= a |y+\rangle_S |\overline{y-}\rangle'_N + b |y-\rangle_S |\overline{y+}\rangle'_N .
\end{aligned} \tag{5.9}$$

In all three cases the subset I, which defines the codewords of the graph code, is the neighbourhood of the spin $N_G(S)$.

In conclusion, after generating N-photons $G(V_{all}, E)$ is initially the star graph. Hence, by only applying $\hat{U}^{LC}$ after generating the last photon we are restricted to graph codes of subgraphs from the LC-equivalence class of the star graph. To reach different LC-equivalence classes, we would need to generate photons after applying $\hat{U}^{LC}$ on the spin (as in section 3.4). However, this requires commuting control phases between spin and photons with the Pauli operators on the spin, which leaves traces of Pauli operators on the newly generated photons. Subsequently, this disturbs the codewords of the graph code and one could not in general write the full state as a state with perfect correlations between spin and graph code.

5.2.2 First search with depolarizing noise

As a starting point, we study a simplified scenario where the entangling of the spin and the graph codes are performed without errors and then the graph code is subjected to depolarizing noise[3]. This allows us to work with graph codes represented by qubits and pure states, which is in contrast to the realistic error model and subsequently is not as computationally heavy. Thus, the depolarizing noise let us do an initial search of the error-correcting capabilities of the codes before applying the realistic error model. The density matrix of a single qubit system subjected to the depolarizing noise transforms as

$$\Lambda(\hat{\rho}) = (1-\epsilon)\hat{\rho} + \frac{\epsilon}{3}(\hat{X}\hat{\rho}\hat{X} + \hat{Y}\hat{\rho}\hat{Y} + \hat{Z}\hat{\rho}\hat{Z}), \tag{5.10}$$

where ϵ is the probability of an error. In other words, the depolarizing noise applies one of the Pauli operators to the qubit with probability $\frac{\epsilon}{3}$, and is readily generalised

[3] No noise applied to the spin for simplicity, since now measuring the logical operator of the code correctly corresponds to indirectly measuring the spin correctly.

to multi-qubit systems. We will call the different graph codes: Star-spin-middle (SSM), Fully-connected (FC) and Star-spin-leaf (SSL). The entanglement between the spin and the graphs codes for three and four photons are illustrated in figure 5.2, and from section 5.2.1 it is straightforward to show that the composite state of the spin and the different graph codes takes the shape:

$$|\Psi\rangle = a\,|0\rangle_S\,|\overline{0}\rangle_{SSM} + b\,|1\rangle_S\,|\overline{1}\rangle_{SSM} = a\,|0\rangle_S \prod_j^N |+\rangle + b\,|1\rangle_S \prod_j^N |-\rangle\,, \tag{5.11}$$

$$|\Psi\rangle = b\,|+\rangle_S\,|\overline{+}\rangle_{SSL} + a\,|-\rangle_S\,|\overline{-}\rangle_{SSL} = b\,|+\rangle_S \prod_j^{N-1} |+\rangle_j\,|0\rangle_\alpha + a\,|-\rangle_S \prod_j^{N-1} |-\rangle_j\,|1\rangle_\alpha\,, \tag{5.12}$$

$$|\Psi\rangle = b\,|y+\rangle_S\,|\overline{y-}\rangle_{FC} + a\,|y-\rangle_S\,|\overline{y+}\rangle_{FC} = b\,|y+\rangle_S \prod_j^N |y+\rangle_j + a\,|y-\rangle_S \prod_j^N |y-\rangle_j\,. \tag{5.13}$$

In (5.12) qubit α is the photon which U^{LC} is applied on to go from FC to SSL. Given that the three different graph codes are all repetition codes they all have the same error-correcting capabilities for their respective measurement basis (with the depolarize noise). Figure 5.3 displays the logical error against probability of a physical error for up to 11 photons. As expected, the larger the graph (i.e. more photons) the more robust the graph code is to errors. Three photons is the lowest number of photons to achieve a lower error rate than direct transmission. Furthermore, the symmetric shape of the graph codes in figure 5.3 is due to the maximum likelihood in the error-correction.

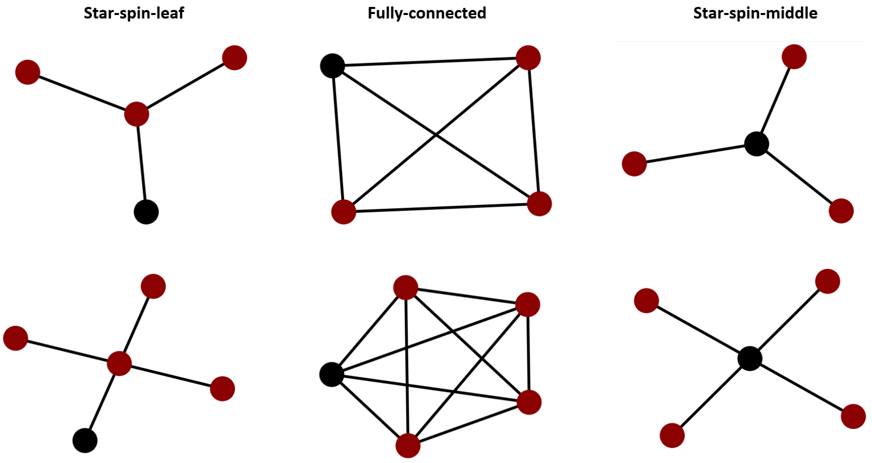


Figure 5.2: The figure shows the spin in black and photons in dark red. The graph objects are the graphs $G(V_{all}, E)$ mentioned in section 5.2.1.

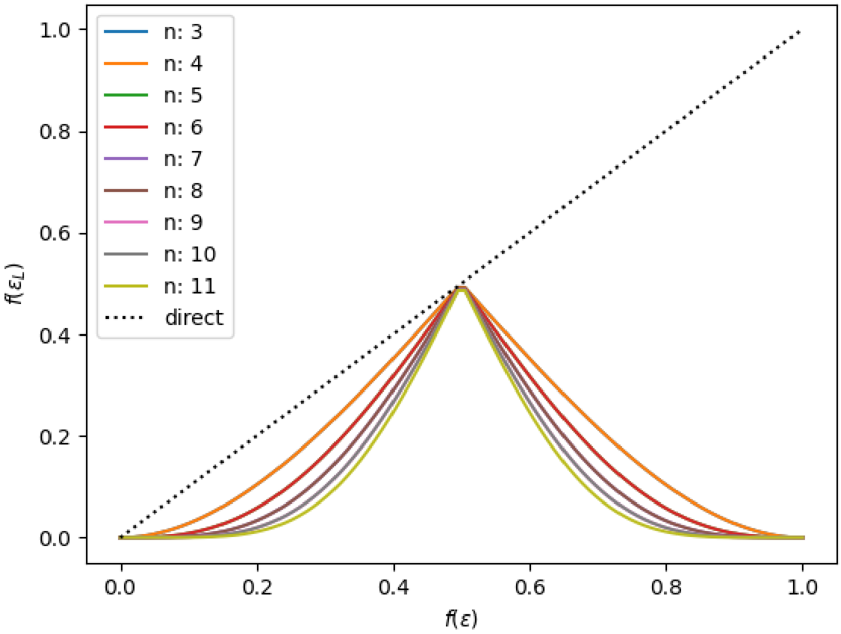


Figure 5.3: The figure shows a plot of the logical error $f(\epsilon_L)$ against the physical error probability $f(\epsilon)$. The different colours indicate the number of photons (n) in the graph codes, and the dotted line direct transmission. Note that $f(\epsilon_L)$ and $f(\epsilon)$ are re-scaled by a factor of $\frac{3}{2}$ against the depolarizing noise error rate ϵ.

5.2.3 Results from the realistic error model

Having constructed a decoder and tested the different graph codes against depolarizing noise, we move on to the realistic error model presented in section 3.3[4]. Given all the error mechanisms in the generation of photons and subsequently measuring them, we investigate if graph codes can outperform single photon entanglement for reading out the spin. We study the different graph codes for three photons, as this was the lowest number of photons to achieve a lower error rate than a single photon.

Before moving on to show results, it is worth highlighting the decoding process with the realistic error model. The process works by first generating the density matrix and then performing projective measurements following the measurement pattern. We project on every possible outcome, building a tree as in figure 5.4 with nodes corresponding to the probability of the measurement outcome. From the tree, measurements can be sampled to obtain $p(S(N))$ and $p(\epsilon_L|S(N))$ in (5.3). At each step (i.e. generating a density matrix for a given set of parameters), we sample 20,000 measurement sequences (i.e. going from top to bottom of the tree) from which we use to calculate the logical error and perform error-correction. Note that the time consuming part of the simulation is not sampling from the tree, but generating the density matrix and the corresponding tree.

[4] Note that now the spin is not free from noise. In the simulations, we have to measure the spin as well to check that it is indeed in the correct state, which now dictates our logical error.

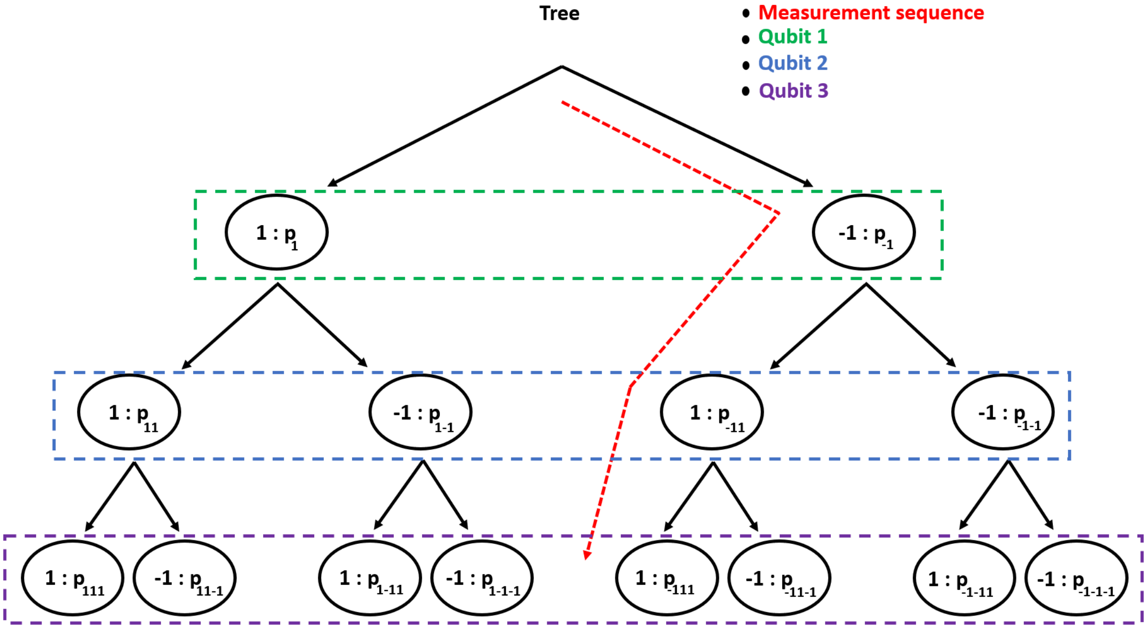


Figure 5.4: The figure illustrates a tree, where going left corresponds to a measurement outcome of 1, and going right corresponds to an measurement outcome of -1. In the figure $\boldsymbol{p}_{ijk}$ represents the probability of measuring qubit i up $(i = 1)$ or down $(i = -1)$ given the previous measurements of qubit j and k.

Since rotation errors are the leading source of errors [30], we start by examining the logical error rate against κ and T_2^* while turning off all other errors. As in section 4.1.3, the density matrix is sampled with 1000 trajectories. From figure 5.5 we observe that the SSL and FC codes achieve a lower logical error rate than direct transmission, while SSM does not give an advantage. We have two hypothesis to the performance of SSL and FC and not SSM. First: As both SSL and FC include one $\hat{U}^{LC}$ on the spin, the $\hat{R}(\frac{\pi}{2})$ from $\hat{U}^{LC}$ can be concatenated with the last $\hat{R}(\pi)$ from the last photon generation to $\hat{R}(\frac{-\pi}{2})$ (as in section 4.1.3). This results in an overall $\hat{R}(\frac{\pi}{2})$ less in their generation, and thus less rotation errors. Second: Rotation errors from κ and T_2^* could induce a certain type of error (e.g. bit flips or phase flips), which SSL and FC can correct well while SSM can not.

The analysis continues by scanning ϵ_L against θ while all other error parameters are set to their values in table 3.1. This is due to θ not having a well characterised experimental value while the other error parameters do. Looking at figure 5.6, we see, as expected from figure 5.5, that an indirect measurement using SSL and FC in a realistic experimental setting beats direct transmission. Furthermore, as θ increases SSM seems to beat direct transmission as well, however, with a larger ϵ_L than SSL and FC.

Moving on, we can examine how the logical error rate scales with photon loss. In figure 5.7 we scan the logical error rate against the probability of photon loss for the three different codes. We have set the experimental parameters to the values given in table 3.1 and $\theta = 0.1°$. Direct transmission is a straight line, since every time the photon is lost we discard the run. As expected, ϵ_L grows with the probability of photon loss p_{loss} as

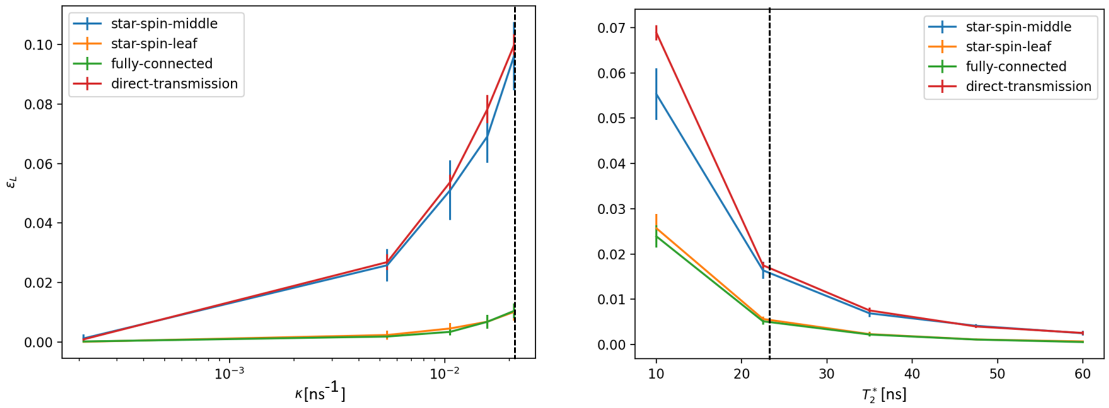


Figure 5.5: The figures shows the logical error rate ϵ_L against the experimental parameters κ and T_2^*, respectively. The different colours indicate different graph codes or direct transmission. Furthermore, the striped line indicate the realistic experimental value for respective parameter.

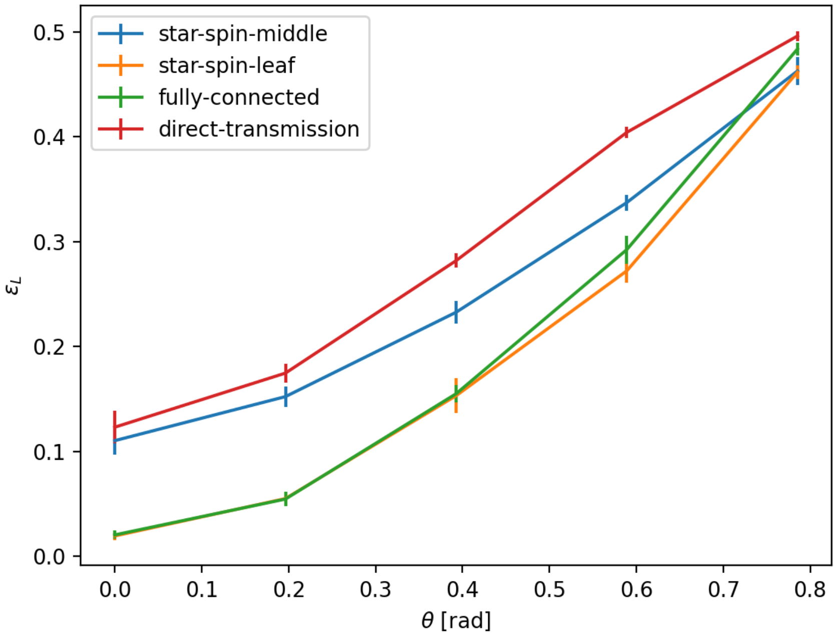


Figure 5.6: The figures shows the logical error rate (ϵ_L) against the angle (θ), which accounts for the imperfect measurement of photons. In the plot, all experimental parameters are set to their realistic values as per table 3.1.

less stabilizers are available. Furthermore, all graph codes outperform direct transmission until a certain threshold value. The threshold value FC is around $P_{loss} \sim 34 - 36\%$, for SSL around $\sim 42 - 44\%$, while for SSM it is around $P_{loss} \sim 3 - 4\%$. We chose to stop at $p_{loss} = 0.7$ since at high P_{loss} no measurements can be performed.

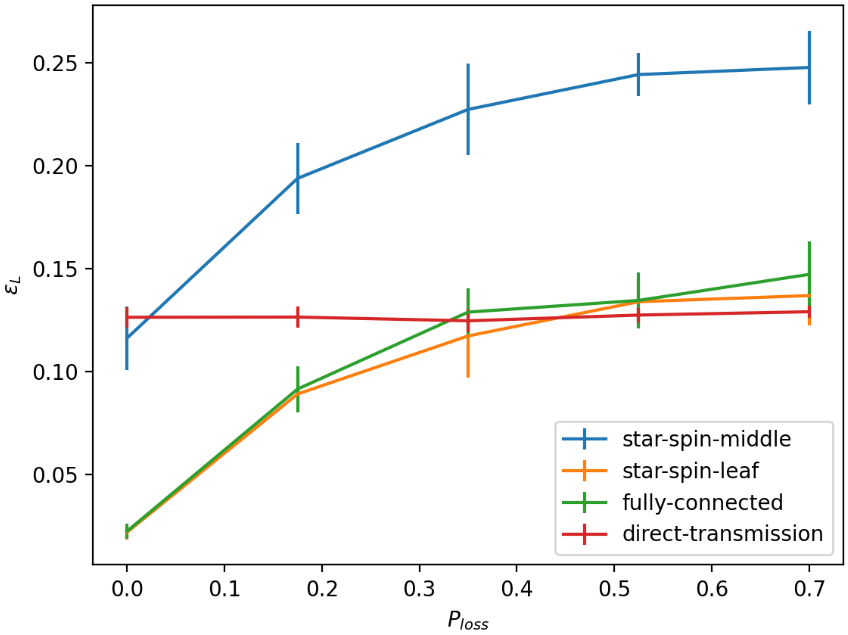


Figure 5.7: The figure shows the logical error rate ϵ_L against photon loss P_{loss}. In the plot, all experimental parameters are set to their realistic values as per table 3.1, and the measurement parameter is set to $\theta = 0.1°$.

5.3 Loss-tolerant and error-corrected quantum key distribution

In this section we study a more practical application of our error and loss protected logical measurements. More specifically, we investigate a novel BB84-protocol for quantum key distribution (QKD) using graph codes instead of single photons. We will focus on graph codes defined by graph states which can be generated by our QD, i.e. the graphs explored in section 3.4. Furthermore, we will want to correct for photon loss and for errors arising in the generation of the graph states.

5.3.1 BB84 with graph codes

The BB84 protocol, first developed in Ref. [54], is a protocol to distribute a secret key between two parties, which is to be used for one time pad encryption. BB84 relies on the concept of mutually unbiased bases (MUB). Two dimensional bases $B_1 = \{B_{1,0}, B_{1,1}\}$ and

$B_2 = \{B_{2,0}, B_{2,1}\}$ are MUB if

$$|\langle B_{1,i}|B_{2,j}\rangle|^2 = \frac{1}{2} \quad \forall i,j \in \{0,1\}. \tag{5.14}$$

That is, if a photon is prepared in one of the two states in B_1 and we try to measure the photon in B_2, both outcomes are equally probable. With MUB, a communication channel (optical fiber or free space) and an encoding for the photons (time-bin, polarization etc.) the BB84 protocol follows:

1. Alice randomly chooses a basis and randomly prepares a photon in either up or down in this basis. In both bases, up is encoded as 0 and down as 1.
2. Alice sends the photon to Bob via the communication channel. Subsequently, Bob randomly chooses a basis which he measures the photon in. If Bob picks the same basis as Alice the outcome is given. However, if Bob chooses the second basis the outcome is random as per MUB.
3. Repeat step 1 and 2 many times and then compare bases of each sent photon via classical communication. All instances where Bob chooses the correct basis, the outcome is correlated with Alice and from these events they can build a secret key (e.g. key=01110). This is illustrated in table 5.1.

Table 5.1: The table illustrates the BB84 protocol for nine rounds using the X and Z bases. In the protocol, up (↑) encodes bit 0 and down (↓) encodes bit 1.

Alice's basis	X	Z	Z	Z	X	X	Z	X	Z
Alice's state	↑	↑	↓	↑	↓	↓	↓	↑	↓
Bob's basis	X	Z	X	X	Z	X	Z	Z	Z
Bob's measurement	↑	↑	↑	↓	↓	↓	↓	↑	↓
Comparing bases	✔	✔	✖	✖	✖	✔	✔	✖	✔
Secret key	0	0				1	1		1

As the photons travel through the communication channel Eve can eavesdrop. A naive attacking strategy for Eve to employ is to simply measure the photons in one of the two bases (B_1 and B_2). After the measurement, Eve prepares a new photon in the measurement outcome and sends it to Bob. However, by comparing the outcome of the first few percent of the rounds (as shown in table 5.1), Alice and Bob can detect if Eve has been eavesdropping. That is, through MUB, eavesdropping can be detected and BB84 presents a secure way of distributing a secret key.

We propose a BB84-protocol using graph codes instead of single photons. As information is redundantly encoded, photon loss and errors, up to a threshold, can be tolerated. The scheme is illustrated in figure 5.8 and works as follows:

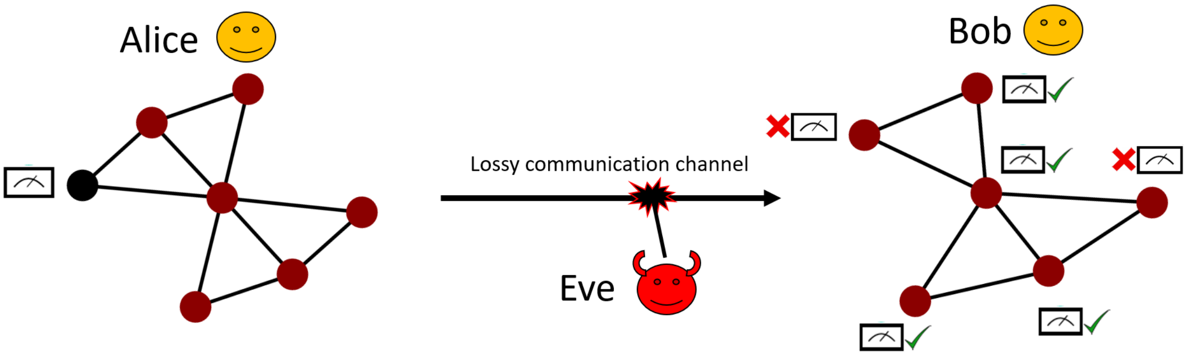


Figure 5.8: The figure illustrates Alice generating a graph state and then measuring the spin, which is represented by the black vertex. Subsequently, Alice sends the projected graph code through a communication channel, which Eve can intercept. Finally the graph code is measured by Bob (illustrated by the graph object of the code), with green marks indicating a successfully measured photon and red cross a lost photon.

1. Alice generates the full graph state associated with $G(V_N, E)$ using her QD. Alice subsequently choose, at random, one of the two bases B_1 or B_2 which she measures the spin in. The measurement projects the state of the measurement outcome to the graph code defined by the subgraph $G[V_N - S] \subseteq G(V_N, E)$ and the subset I being $N_G(S)$ (i.e. the neighbours of the spin). This follows the second approach of encoding a physical qubit in a graph code discussed in section 2.2.2.

2. Alice sends the graph code to Bob, and Bob randomly chooses a basis to measure the code in, i.e. one of the two logical bases $\overline{B_1}$ or $\overline{B_2}$. Again, if Bob picks the same basis as Alice the outcome is correlated, but if he chooses the other basis the outcome is random as per MUB.

3. Repeat steps 1 and 2 many times to generate the secret key.

Generating one photon in this protocol is equivalent to initialising the spin in $|+\rangle_S$, do one round of the time-bin entanglement protocol and measure the spin. The motivation for the graph code protocol is that the time scale of initialising and measuring the spin is much longer than the time scale ($\sim$ 400 ns) of generating time-bin entangled photons ($\sim$ 20 ns) [30]. Thus, we assume the rate at which information is sent between Alice and Bob is not affected by the additional time it takes to generate more photons. Furthermore, an attractive feature of this protocol is the true randomness in the choice of state sent by Alice, induced from the measurement.

5.3.2 Secret key rate

A common metric used to determine the performance of any QKD protocol is the secret key rate generated between Alice and Bob. In the case of Alice and Bob using privacy amplification and discarding rounds where they measure in different bases, a lower bound on the secret key rate is given by [55]

$$R \geq T(I_{AB} - \mathbf{min}(I_{AE}, I_{BE})), \tag{5.15}$$

where T stands for the transmission, I_{AB} for the mutual information between Alice and Bob, and I_{AE}/I_{BE} the mutual information between Alice/Bob and Eve. However, as in Ref. [55] only I_{AE} is considered. The mutual information between Alice and Eve is dependent on the attacking strategy Eve uses. Ultimately, the information Eve acquires scales with the disturbance she induces, which makes it more likely for Bob and Alice to detect her resulting in a zero key rate. To have a non zero key rate, Bob must have more information than Eve $I_{AB} > I_{AE}$.

We will use an expression for the secret key rate, where it is assumed that Eve performs symmetric attacks [56, 57], in which case the secret key rate is given by

$$R \geq T(Flog_2(F) + (1-F)log_2(1-F) - F_e log_2(F_e) - (1-F_e)log_2(1-F_e)), \tag{5.16}$$

where F is the fidelity of the state sent between Alice and Bob, and $F_e = \frac{1+2\sqrt{F(1-F)}}{2}$ is the optimal fidelity for Eve given her symmetric attacks [57, 55]. That is, the secret key rate is a function of transmission and the fidelity of the state sent to Bob. In the graph code scheme, we will define the fidelity F as the average fidelity of measuring the graph code in the two MUB logical bases

$$F = \langle 1 - \epsilon_L^{\overline{B_1}}, 1 - \epsilon_L^{\overline{B_2}} \rangle. \tag{5.17}$$

Furthermore, the transmission will be defined as the average success probability for Bob to measure the graph code in both logical bases

$$T = \langle T^{\overline{B_1}}, T^{\overline{B_2}} \rangle, \tag{5.18}$$

where $T^{\overline{B_1}}$ and $T^{\overline{B_2}}$ stands for the transmission rate of respective logical basis. The motivation for the graph code BB84 scheme is that we can perform error-correction to increase F, and find loss-tolerant graph codes to increase T. In this way, we could potentially achieve a higher secret key rate than what is achievable with a single-qubit.

5.3.3 Loss-tolerant graph codes

Photon loss is arguably the key challenge to reach long distance QKD in optical fiber channels. This is because transmission decreases exponentially $T = e^{\frac{-L}{L_0}}$ with distance L. Therefore, we will begin the study by analysing only the loss-tolerance of the graph codes (i.e. we set $F = 1$), and scan the secret key rate (SKR) over the dimensionless parameter $\frac{L}{L_0}$. Furthermore, we choose the two bases B_1 and B_2 as two different Pauli bases.

We begin by searching for graph codes, which achieves a higher SKR than direct transmission, defined by the graph states found in section 3.4. This resulted in numerous of graph codes, which is summarised in table 5.2. In figures 5.9a and 5.9b we illustrate the graph states and the respective key rates for one of the best graph codes (before measuring the spin) for four-qubits, five-qubits, six-qubits, seven-qubits and eight-qubits.

The SKR versus P_{loss} plot in figure 5.9b was included to better illustrate the loss-tolerance of the graph codes. Although the loss-tolerance increases with the number of qubits, the increase is very small. Furthermore, note in figure 5.9b no graph code has a threshold larger than 50%, which has a simple explanation first stated in Ref. [58]. Imagine the threshold is above 50%, then Alice sends half of the qubits of the graph code to Bob and half to Charlie. Bob and Charlie are both above the threshold and can both measure the graph code with high probability, effectively cloning the state which breaks the no cloning theorem [14].

Table 5.2: Number of loss-tolerant graph codes for four to eight number of qubits.

Number of qubits	4-qubits	5-qubits	6-qubits	7-qubits	8-qubits
Number of graph codes	44	176	704	2816	11264

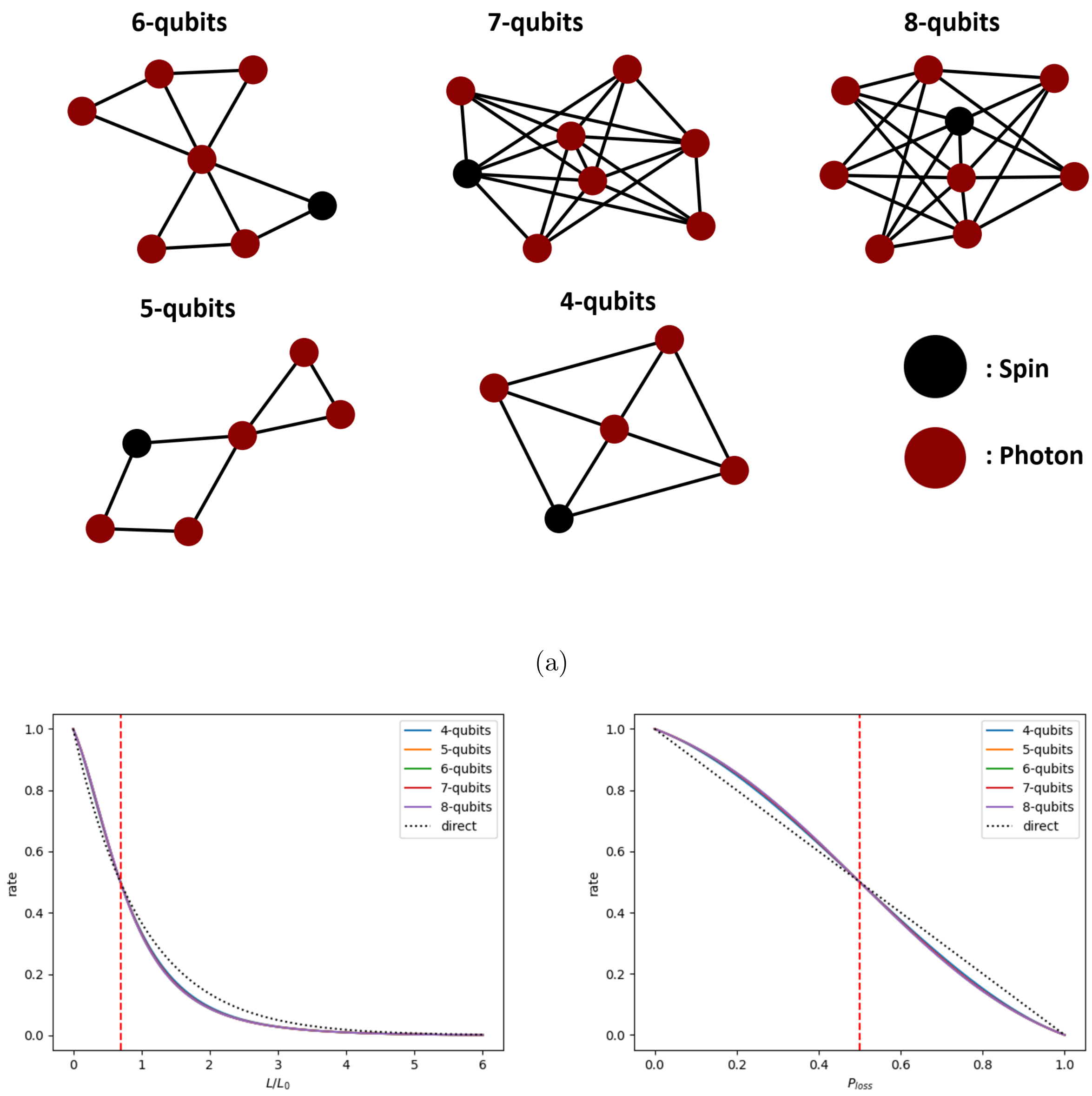
Arguably, we would expect a bigger increase in performance for the larger graph codes in figure 5.9b. We hypothesise that this is due the entanglement structure of the available graph states with a single QD. We do not have a clear argument for this hypothesis, and investigating it might be interesting future work. However, for example, a eight-qubit code that cannot be generated with a single QD which is more loss-tolerant than the best eight-qubit code shown in figure 5.9b is displayed in figure 5.10. This graph code illustrates a clear increase in performance and a more asymptotic behaviour compared to the code in figure 5.9b.

5.3.4 Loss and error-tolerant graph code

A more comprehensive study of the graph code BB84 protocol include both photon loss and gate errors (i.e. $F < 1$ in (5.17)) when calculating the SKR. More specifically, we are interested in errors from the generation of the graph codes themselves, as was done in section 5.2.3. Further, we choose to study the four-qubit codes (five qubits with the spin), which is due to larger codes rendering our error model impractical as the QD and the photons are represented by qudits[5].

As was done in section 5.2.2, we first analyse the codes while subjected to the depolarizing noise. This is due to the real error model being time consuming, and thus the depolarizing noise acts as a first filter to find promising codes. While there are graph codes, which on average have a higher transmission rate in two bases than direct transmission, no such graph codes were found for logical error rates using the depolarizing noise. That is, with an ideal noise channel, we found no graph code which can correct for errors in two different logical bases such that the logical error is lower than that of a single-qubit simultaneously in both bases. With that said, error-correction can still be used to suppress the logical

[5] This was not the case in the search of loss-tolerant codes since photons and the QD could be represented by qubits.



(a)

(b)

Figure 5.9: (a) The graphs of the best graph codes before measuring the spin. (b) The left figure is a plot of the SKR versus L/L_0, while the right is a plot of the SKR versus photon loss P_{loss}. The graph codes are symbolised by their number of qubits and the red striped line indicates 50% photon loss.

error rate.

With the depolarizing noise, a promising four-qubit code for both loss-tolerance and error-correction in logical-X and logical-Z is shown in figure 5.11 along with the logical error rates for respective basis. Not shown in figure 5.11 is the code's loss-tolerance, which is equal to that of the four-qubit code shown in figure 5.9a. Another attractive feature with the code shown in figure 5.11 is that LC is applied to the spin during its generation.

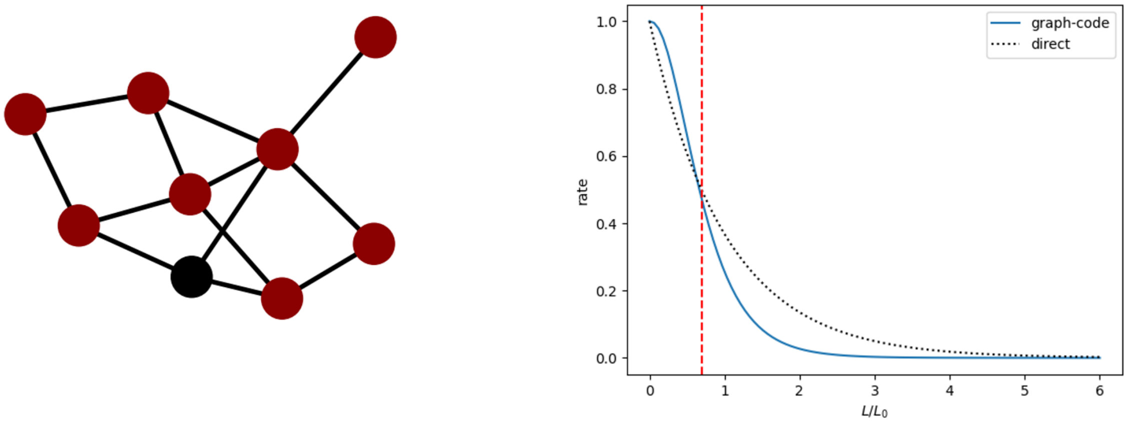


Figure 5.10: The figure illustrates one example of a graph which can not be generated by a single QD. The code defined by the graph and the neighbours of the black vertex show a more asymptotic behaviour in the SKR than those generated by the QD.

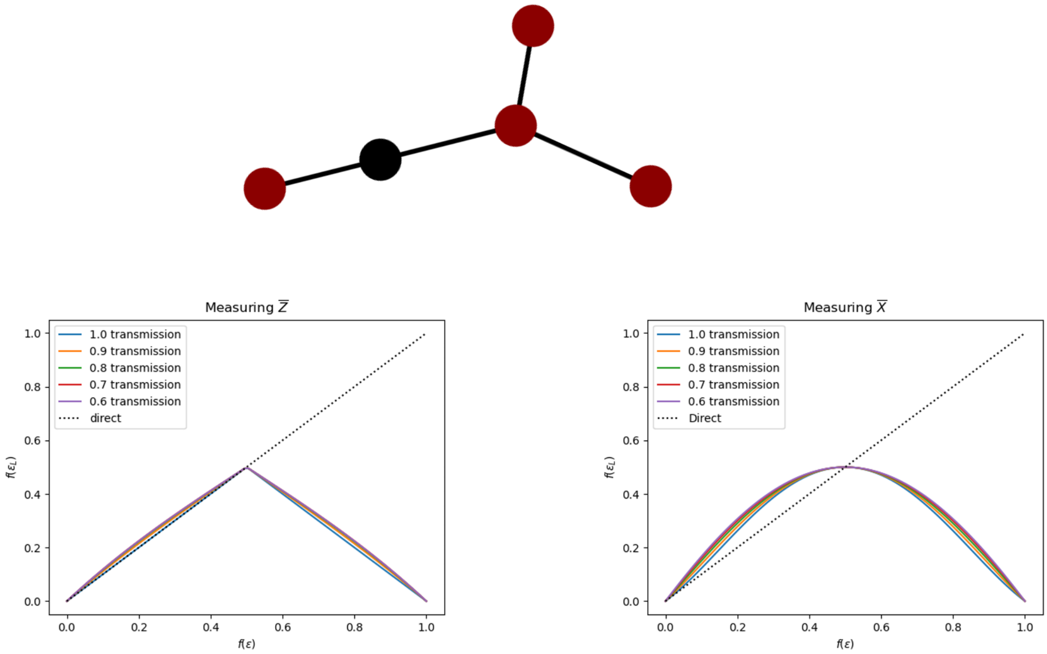


Figure 5.11: The figure illustrates the logical error rate for measuring $\overline{Z}$ and $\overline{X}$ of the code defined by the graph above. This is shown for different transmission rates.

We move on to analysing the code presented in figure 5.11 with the realistic error model. We study two sets of experimental parameter settings. First, all errors turned off except $T_2^* = 23.2$ ns, which represents an ideal case. Second, all parameters set to the values in table 3.1, which presents a more realistic scenario. At each step 100 trajectories are used to sample the density matrix. As seen in the figure 5.12 in an ideal setting, we still manage to beat direct transmission up to $P_{loss} = 50\%$, although with the difference shrunk. Furthermore, note that the key rate is higher for the code than direct transmission at 100% transmission. This is solely due to the error-correcting capability of the code. In a more realistic setting, however, the errors are simply to large and already at $P_{loss} = 10\%$

we get a non positive key rate for the code.

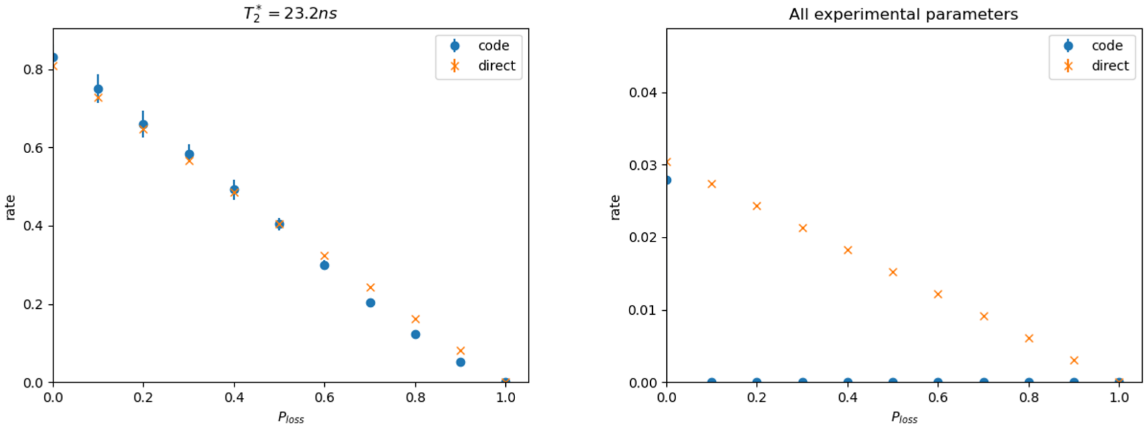


Figure 5.12: The figure illustrates the SKR against photon loss for the code in figure 5.11 with the real error model in the two different settings. The orange crosses represents direct transmission with a single qubit and the blue dots represents the code.

Chapter 6

Conclusion and Outlook

The backbone of this study has been a QD graph state generator. First an exhaustive graph state search was conducted. Second, given the graph states from the graph search, we explored and simulated various near-term applications possible with this device considering a realistic error model obtained by generalising that of Ref. [30] to multi-qubit states. Some conclusions can be drawn from this analysis:

1. The quantum dot, given the time-bin entanglement protocol, is an excellent graph state generator which gives access to a plethora of graph states. Furthermore, the error model accounts for most of the infidelities in the device and is directly linked to experimental parameters, which makes it a great tool for simulations. However, the error model is rather involved with a large Hilbert space, where the QD and the photons are represented as qudits instead of qubits. Furthermore, the Monte-Carlo approach makes it computationally heavy by having to construct a density matrix. Further work could be made in to making the error model fully unitary, abandoning the collapse operators and Monte-Carlo approach, to increase its computational efficiency and to scale to larger graph states. Also, mapping the excitation and decay errors to faulty two-qubit gates would allow to represent the QD and photons as qubits and alleviate the size of the simulation Hilbert space.

2. A measurement based implementation of Grover's search algorithm and Deutsch's algorithm for two logical qubits yielded an average success rate of $\sim 70\%$, which given the current experimental parameters indicates they are error robust. The P-VQE, however, did not perform as well yielding errors around $\sim 30\%$ compared to the exact ground state energies for only a two-qubit experiment. To reach an error less than 1% a $T_2^* = 73.2$ would be needed, with all other errors gone. In future work exploring more noisy robust algorithms, which can be implemented on graph states would be interesting. Furthermore, it would be interesting to see if any quantum chemistry problem could be mapped to the Hamiltonians of any of the available graph states we discovered with our QD graph state generator.

3. In terms of quantum error-correction we discovered two interesting applications of logical measurements, which does not require ancilla qubits. First, we applied this to a scenario of reading out the spin state of our QD, and managed to outperform the baseline case of using one photon. This scenario is interesting as it corrects

for errors that arise from the generation of the code itself. Second, we looked at a QKD scheme and analysed how the secret key rate scaled with photon loss and errors using graph codes. We found codes that yielded a higher secret key rate than a single-qubit when accounting for photon loss and in an ideal setting of low errors. However, as more errors were introduced this changed. The main problem for the error-correction was finding graph codes which could sufficiently correct a logical measurement in two different bases simultaneously. One interesting direction to follow, which connects the two applications considered in this thesis, would be error and loss protected quantum algorithms using graph codes. With the logical measurements fitting perfectly for measurement based quantum computing.

Appendix A

Graph zoology

We here display all graphs (excluding isomorphic graphs) which can be generated by our QD in section 3.4 for up to five vertices. In figure A.1 all the graphs are displayed, grouped in their LC-class indicated by the boxes. Vertices are given the colour blue to better illustrate their labels, with label "0" given to the spin.

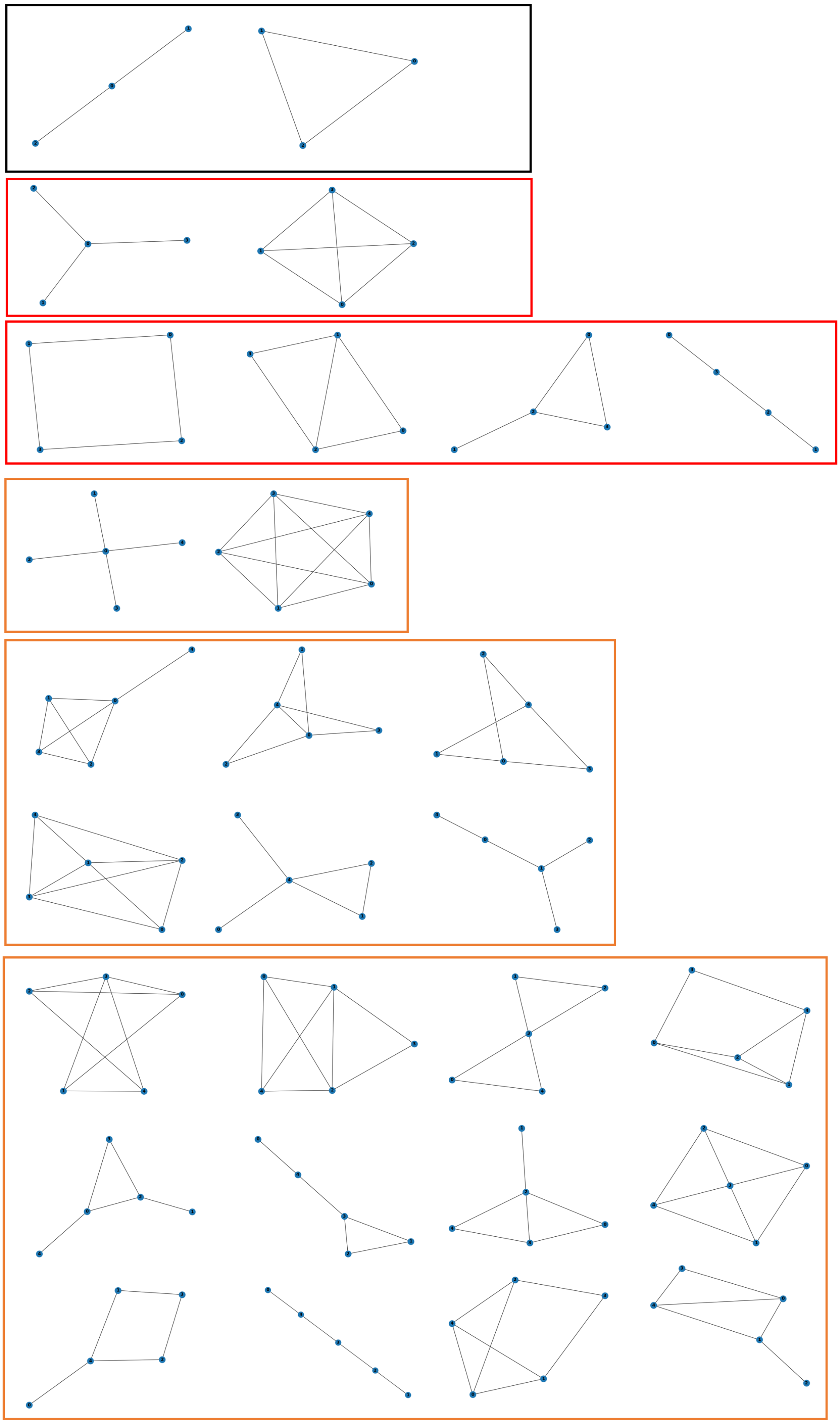


Figure A.1: Graphs which can be generated by a single QD for up five-qubits, where we ignore displaying isomorphic graphs. The boxes indicate a LC-class, and the colours of the boxes indicate the number of vertices: 3-vertices=black, 4-vertices=red and 5-vertices=orange.

Appendix B

Transformation of stabilizers under local Pauli measurements

Just as local complementation maps stabilizer states to stabilizer states, so does Pauli measurements. In this Appendix we will show how stabilizer states transform under Pauli measurements, which is also described in Ref. [17] and derived in Ref. [24]. First off, the transformation of the stabilizer generators under single Pauli measurements follow the transformation rules [17, 24]:

Transformation rules of stabilizer generators under single Pauli measurements: *Consider a stabilizer state defined by its stabilizer generators $\mathcal{S} = \{\hat{g}_n\}$ on which we perform a single Pauli measurement $\hat{P}^{[i]}$ on qubit i. If $\hat{P}^{[i]}$ anticommutes with at least one generator $\{\hat{P}^{[i]}, \hat{g}_k\} = 0$ the generators in $\mathcal{S}$ transforms as:*

1. *If the generator commutes with the measurement $[\hat{g}_j, \hat{P}^{[i]}] = 0$: $\hat{g}_j \rightarrow \hat{g}_j$*

2. *If the generator, different from k, anticommutes with the measurement $\{\hat{g}_i, \hat{P}^{[i]}\} = 0$: $\hat{g}_j \rightarrow \hat{g}_k \hat{g}_j$*

3. *Generator k: $\hat{g}_k \rightarrow \pm \hat{P}^{[i]}$ depending on the outcome of the measurement.*

Note that the choice of generator k is arbitrary. After the measurement qubit i is separable from all the other qubits and in an eigenstate of $\hat{P}^{[i]}$. Furthermore, all generators now commute with $\hat{P}^{[i]}$, meaning that they either act with $\hat{P}^{[i]}$ or $\hat{\mathbb{1}}$ on qubit i [24].

As stabilizers are products of stabilizer generators $\hat{S}_D = \prod_{j \in D} \hat{g}_j$, their transformation follows directly from the transformation of the generators. We will consider $\hat{S}_D$ and a single Pauli measurement $\hat{P}^{[i]}$ with an anticommuting generator $\hat{g}_k$. Furthermore, we define the set $D' = \{j \in D | \{\hat{g}_j, \hat{P}^{[i]}\} = 0\} \subset D$ as the subset of indices in D that anticommutes with $\hat{P}^{[i]}$. Now, $\hat{g}_k$ can be in D or not, and the transformation of $\hat{S}_D$ in the two different cases follow:

1. $\hat{g}_k \notin D$:

$$\hat{S}_D \to \prod_{i\in D/D'} \hat{g}_i \prod_{j\in D'} \hat{g}_k\hat{g}_j = \hat{g}_k^{|D'|}\hat{S}_D = \begin{cases} \hat{S}_D, & \text{if} \quad [\hat{S}_D, \hat{P}^{[i]}] = 0 \\ \hat{g}_k\hat{S}_D, & \text{if} \quad \{\hat{S}_D, \hat{P}^{[i]}\} = 0 \end{cases}$$

2. $\hat{g}_k \in D$:

$$\hat{S}_D \to \pm P^{[i]} \prod_{i\in D/D'} \hat{g}_i \prod_{j\in D'/k} \hat{g}_k\hat{g}_j = \pm P^{[i]}\hat{g}_k^{|D'-2|}\hat{S}_D$$

$$= \begin{cases} \pm P^{[i]}\hat{S}_D, & \text{if} \quad [\hat{S}_D, \hat{P}^{[i]}] = 0 \\ \pm P^{[i]}\hat{g}_k\hat{S}_D, & \text{if} \quad \{\hat{S}_D, \hat{P}^{[i]}\} = 0 \end{cases}$$

In the derivation, we used the fact that if a stabilizer $\hat{S}_D$ commutes or anticommutes with $\hat{P}^{[i]}$ there are an even or odd number of indices in D, respectively. The significance of this derivation is that if a stabilizer $\hat{S}_D$ commutes with $\hat{P}^{[i]}$, the local Pauli operators on all qubits are left unchanged except on qubit i, which has to be $\hat{P}^{[i]}$ or $\hat{\mathbb{1}}$ [24]. Note that this generalises to graph states as they are stabilizer states.

Appendix C

Decoding algorithm

Algorithm 2 Loss-tolerant and error-corrected decoding

```
First step: Find all measurement patterns
$\mathcal{M} = \emptyset$    ▷ Initialise list of measurement patterns.
$\Theta \leftarrow$ FindAll($\mathcal{T}$, $L$)    ▷ Given graph code $\mathcal{T}$, FindAll generates all measurement patterns for compatible with logical operator $L$.
for $m_i \in \Theta$ do
    ${\mathcal{S}_{\mathcal{R}}}^i \leftarrow$ FindS($m_i$)    ▷ FindS finds all stabilizers compatible with $m_i$.
    Append $\{m_i, {\mathcal{S}_{\mathcal{R}}}^i\}$ to $\mathcal{M}$    ▷ Add the measurement pattern to $\mathcal{M}$.
end for
Second step: Run LT-EC decoder
$\mathcal{M}_{\mathcal{R}} = \emptyset$    ▷ Initialise list with measurement outcomes.
while Measurement active do
    $\{m_i, {\mathcal{S}_{\mathcal{R}}}^i\} \leftarrow$ FindBest($\mathcal{M}$)    ▷ FindBest finds the best pattern, which is the one with $m_i$ of lowest weight and largest $|\mathcal{S}_{\mathcal{R}}|$.
    $\sigma_k \leftarrow \{0, 1, na\}$    ▷ Try measuring qubit $k$, which, if qubit is not lost, gives a value 0 or 1 and if lost $na$.
    $\mathcal{M}_{\mathcal{R}} \leftarrow \mathcal{M}_{\mathcal{R}} + \sigma_k$    ▷ Append measurement outcome to list $\mathcal{M}_{\mathcal{R}}$.
    if $\sigma_k = na$ then    ▷ Update measurement strategies $\mathcal{M}$ if qubit $k$ was lost or not.
        $\mathcal{M} \leftarrow \mathcal{M} - \{M_k\}$    ▷ Remove all measurement patterns $\{M_k\}$ with a Pauli operator different from identity on qubit $k$.
    else
        $\mathcal{M} \leftarrow \mathcal{M}$
    end if
end while
if $\mathcal{M}_{\mathcal{R}} = \emptyset$ then    ▷ If no measurement of $L$ could be performed the logical qubit is lost. Otherwise, run the EC decoder with the measurement outcomes in $\mathcal{M}_{\mathcal{R}}$.
    Logical qubit lost.
else
    Run EC decoder with $\mathcal{M}_{\mathcal{R}}$
end if
```

Bibliography

[1] J. Preskill, "Introduction and overview," 2020.

[2] J. Johansson, P. Nation, and F. Nori, "Qutip 2: A python framework for the dynamics of open quantum systems," *Computer Physics Communications*, vol. 184, no. 4, pp. 1234–1240, 2013.

[3] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics.* Cambridge University Press, 2 ed., 2017.

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* USA: Cambridge University Press, 10th ed., 2011.

[5] D. P. DiVincenzo, "Two-bit gates are universal for quantum computation," *Phys. Rev. A*, vol. 51, pp. 1015–1022, Feb 1995.

[6] David, R. Deutsch, and Jozsa, "Rapid solution of problems by quantum computation," *Royal Society*, vol. 439, Dec 1992.

[7] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–328, Jul 1997.

[8] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, p. 4213, Jul 2014.

[9] J. Preskill, "Quantum algorithms," 2020.

[10] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, aug 2018.

[11] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. Booth, and J. Tennyson, "The variational quantum eigensolver: a review of methods and best practices," 11 2021.

[12] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New Journal of Physics*, vol. 18, p. 023023, feb 2016.

[13] S. Paesani, *Large-scale integrated quantum photonics: Development and applications.* PhD thesis, University of Bristol, 2019.

[14] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802–803, Oct 1982.

[15] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493–R2496, Oct 1995.

[16] C. Vigliar, *Graph states in large-scale integrated quantum photonics.* PhD thesis, University of Bristol, 2020.

[17] D. Gottesman, *Stabilizer Codes and Quantum Error Correction.* PhD thesis, California Institute of Technology, 1997.

[18] E. Knill, R. Laflamme, and L. Viola, "Theory of quantum error correction for general noise," *Physical Review Letters*, vol. 84, pp. 2525–2528, mar 2000.

[19] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," *Reports on Progress in Physics*, vol. 76, p. 076001, Jun 2013.

[20] M. Van den Nest, J. Dehaene, and B. De Moor, "Efficient algorithm to recognize the local clifford equivalence of graph states," *Phys. Rev. A*, vol. 70, p. 034302, Sep 2004.

[21] D. Schlingemann, "Stabilizer codes can be realized as graph codes," 2001.

[22] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. Nest, and H. Briegel, "Entanglement in graph states and its applications," vol. 162, 03 2006.

[23] M. Van den Nest, J. Dehaene, and B. De Moor, "Graphical description of the action of local clifford transformations on graph states," *Physical Review A*, vol. 69, Feb 2004.

[24] S. Paesani, T. Bell, A. Jones, and L. Pettersson, "Notes on optimising loss-tolerance in stabilizer codes for photonic qubits," 2021.

[25] S. Paesani and C. Vigliar, "Notes on graph codes," 2021.

[26] N. H. Lindner and T. Rudolph, "Proposal for pulsed on-demand sources of photonic cluster state strings," *Phys. Rev. Lett.*, vol. 103, p. 113602, Sep 2009.

[27] M. Appel, A. Tiranov, A. Javadi, M. Löbl, W. Ying, S. Scholz, A. Wieck, A. Ludwig, R. Warburton, and P. Lodahl, "A coherent spin-photon interface with waveguide induced cycling transitions," 06 2020.

[28] R. Uppu, L. Midolo, X. Zhou, J. Carolan, and P. Lodahl, "Quantum-dot-based deterministic photon–emitter interfaces for scalable photonic quantum technology," *Nature Nanotechnology*, vol. 16, pp. 1308–1317, Dec 2021.

[29] P. Lodahl, S. Mahmoodian, and S. Stobbe, "Interfacing single photons and single quantum dots with photonic nanostructures," *Rev. Mod. Phys.*, vol. 87, pp. 347–400, May 2015.

[30] M. Hayhurst, Appel, *A Quantum Dot Source of Time-Bin Multi-Photon Entanglement.* PhD thesis, University of Copenhagen, 2021.

[31] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond bell's theorem," 2007.

[32] K. Tiurev, P. L. Mirambell, M. B. Lauritzen, M. H. Appel, A. Tiranov, P. Lodahl, and A. S. Sørensen, "Fidelity of time-bin-entangled multiphoton states from a quantum emitter," *Physical Review A*, vol. 104, nov 2021.

[33] J. H. Bodey, R. Stockill, E. V. Denning, D. A. Gangloff, G. Éthier Majcher, D. M. Jackson, E. Clarke, M. Hugues, C. L. Gall, and M. Atatüre, "Optical spin locking of a solid-state qubit," *npj Quantum Information*, vol. 5, Nov 2019.

[34] M. Arcari, I. Söllner, A. Javadi, S. Lindskov Hansen, S. Mahmoodian, J. Liu, H. Thyrrestrup, E. H. Lee, J. D. Song, S. Stobbe, and P. Lodahl, "Near-unity coupling efficiency of a quantum emitter to a photonic crystal waveguide," *Phys. Rev. Lett.*, vol. 113, p. 093603, Aug 2014.

[35] A. Dahlberg, J. Helsen, and S. Wehner, "Counting single-qubit clifford equivalent graph states is p-complete," *Journal of Mathematical Physics*, vol. 61, p. 022202, Feb 2020.

[36] sammorley short, "sammorley-short/gsc: v2.0," Mar. 2019.

[37] J. Craig, Adcock, *Generating optical graph states.* PhD thesis, University of Bristol, 2019.

[38] R. Raussendorf and H. J. Briegel, "Quantum computing via measurements only," *arXiv:quant-ph/0010033*, Oct 2000.

[39] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, Jan 2018.

[40] S. Bartolucci, P. Birchall, H. Bombin, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kieling, N. Nickerson, M. Pant, F. Pastawski, T. Rudolph, and C. Sparrow, "Fusion-based quantum computation," 2021.

[41] R. Raussendorf and H. J. Briegel, "A one-way quantum computer," *Phys. Rev. Lett.*, vol. 86, pp. 5188–5191, May 2001.

[42] R. Jozsa, "An introduction to measurement based quantum computation," *Quantum Information Processing*, vol. 199, 09 2005.

[43] M. Newman, L. A. de Castro, and K. R. Brown, "Generating Fault-Tolerant Cluster States from Crystal Structures," *Quantum*, vol. 4, p. 295, July 2020.

[44] R. Raussendorf, D. E. Browne, and H. J. Briegel, "Measurement-based quantum computation on cluster states," *Phys. Rev. A*, vol. 68, p. 022312, Aug 2003.

[45] G. Vallone, E. Pomarico, F. De Martini, and P. Mataloni, "One-way quantum computation with two-photon multiqubit cluster states," *Physical Review A*, vol. 78, Oct 2008.

[46] . L. D. W. D. A. Chan, Z. Shi and C. A. Muschik, "A measurement-based variational quantum eigensolver," 11 2022.

[47] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[48] C. Tsallis and D. A. Stariolo, “Generalized simulated annealing,” *Physica A: Statistical Mechanics and its Applications*, vol. 233, no. 1, pp. 395–406, 1996.

[49] D. J. Wales and J. P. K. Doye, “Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms,” *The Journal of Physical Chemistry A*, vol. 101, pp. 5111–5116, Jul 1997.

[50] P. Sarvepalli and R. Raussendorf, “Efficient decoding of topological color codes,” *Physical Review A*, vol. 85, Feb 2012.

[51] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, “Towards practical classical processing for the surface code,” *Physical Review Letters*, vol. 108, May 2012.

[52] H. Bombin, R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, “Strong resilience of topological codes to depolarization,” *Physical Review X*, vol. 2, Apr 2012.

[53] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, “Optimizing quantum error correction codes with reinforcement learning,” *Quantum*, vol. 3, p. 215, Dec 2019.

[54] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” *Theoretical Computer Science*, vol. 560, pp. 7–11, 2014. Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84.

[55] Y. Ding, D. Bacco, K. Dalgaard, X. Cai, X. Zhou, K. Rottwitt, and L. K. Oxenløwe, “High-dimensional quantum key distribution based on multicore fiber using silicon photonic integrated circuits,” *npj Quantum Information*, vol. 3, p. 25, Jun 2017.

[56] M. Arpita and P. Goutam, “Another look at symmetric incoherent optimal eavesdropping against bb84,” 2011.

[57] D. Bruß, “Optimal eavesdropping in quantum cryptography with six states,” *Phys. Rev. Lett.*, vol. 81, pp. 3018–3021, Oct 1998.

[58] M. Varnava, D. E. Browne, and T. Rudolph, “Loss tolerance in one-way quantum computation via counterfactual error correction,” *Phys. Rev. Lett.*, vol. 97, p. 120501, Sep 2006.