

Distributed quantum memory using NV centers

Eduardo Villaseñor



August 26, 2018

Delft University of Technology

Faculty of Applied Sciences
QuTech

MSc. thesis.

Distributed quantum memory using NV centers

in order to obtain the degree of MSc. in Applied Physics by

Eduardo Villaseñor

1. Reviewer **Tim Taminiau**
QuTech
Delft University of Technology

2. Reviewer **Barbara Terhal**
QuTech
Delft University of Technology

Supervisor **David Elkouss**

August 26, 2018

Eduardo Villaseñor

Distributed quantum memory using NV centers

MSc. thesis., August 26, 2018

Reviewers: Tim Taminiau and Barbara Terhal

Supervisor: David Elkouss

Delft University of Technology

QuTech

Faculty of Applied Sciences

Mekelweg 2

2628 CD Delft

Abstract

There are still great hurdles to overcome in the construction of a practical quantum computer. Most significantly, there is still great need for less noisy operation devices and the ability to scale the computer into hundreds or thousands of qubits. In this work we study a approach to construct a practical quantum computer which is fundamentally scalable. We consider Nitrogen vacancy centers as nodes in a quantum network to model a distributed surface code. The analysis consists first in a study of how the surface code can be implemented over a quantum network and which error models we consider to get a realistic representation of the system. Thereafter, we calculate the error thresholds through the parameters of the most significant error sources in order to determine the code effectiveness. After the initial implementation we explore further designs which by utilizing additional resources in the network aim to reduce the impact of noise in the system. The results show a considerable improvement over the initial implementation and provide some insights into further developments that could produce better results.

Acknowledgement

Hereby I would like to acknowledge the support of many people through this work and in the path that has led me here. To my father, who planted the seed of curiosity and showed me how science can be more than a profession but also serve as a great source of fulfillment. To my mother, who motivated me to follow my own ideas and dreams above anything else. All while teaching me the importance of never taking anything at face value but to challenge everything, specially my own ideas. To my advisors David and Tim who where patient with my mistakes and guided me through this process with very insightful discussions. And finally to my friends, whom shared with me equally my happiness as well as my frustrations. Without their support this would have never been possible.

Contents

1	Introduction	1
2	Fault tolerant quantum computing	5
2.1	Concepts of Quantum Computing	5
2.1.1	Density matrices	6
2.1.2	Channels	7
2.1.3	Circuits	7
2.2	Quantum error correction	8
2.3	Surface code	10
2.3.1	Faulty measurements syndrome extraction	10
2.3.2	Syndrome decoding	12
2.3.3	Thresholds	13
2.3.4	Numerical simulations	14
3	NV centers as quantum nodes	19
3.1	NV centers in quantum computing	19
3.1.1	Hardware limitations	20
3.2	Error models	21
3.3	NV centers in a network	22
3.3.1	Entanglement generation	23
3.3.2	Barret-Kok	26
3.3.3	Extreme photon loss	26
3.4	Parameters	27
3.5	Outline	28
4	Distributed surface code	29
4.1	Distributed stabilizer measurements	29
4.2	GHZ state generation protocols	32
4.2.1	Dealing with indeterministic state generation	33
4.2.2	Stabilizer measurement scheduling	35
4.2.3	Additional protocols: Entanglement purification	35
4.3	Simulating a quantum network	36

4.3.1	Protocols comparison	38
4.4	Effectiveness of the distributed implementation	39
4.5	Outline	42
5	Expanding network resources	43
5.1	Parallel entanglement generation	43
5.2	Clustering data qubits	46
5.3	Outline	52
6	Conclusions	53
A	Appendix A: Definitions and derivations	55
A.1	Fidelity	55
A.2	State twirling	55
A.3	GHZ state generation	56
A.4	Parametrization over the parameter space	57
B	Appendix: Noise modeling	59
B.1	The χ -matrix	59
B.1.1	The Choi-Jamiolkowski isomorphism	60
B.2	Noisy stabilizer modeling	61
	Bibliography	65

Introduction

The idea of a quantum computer emerged accompanied with both the realizations of the great potential of this new technology and the immense difficulty involved in creating a practical large scale quantum computer. Due to the nature of the quantum scale, a quantum computer will inevitably suffer from the presence of noise at the most fundamental level. To fight this noise, besides the great attempts to engineer low noise inducing devices, we strive to use quantum resources into creating large systems which are capable of error correction. Analogously to doing error correction on a classical computer, in quantum error correction we make use of redundancies of the system provided by the extra resources. The idea is to cleverly store the quantum information in a subsystem in such a way that we can perform measurements on the larger system in order to detect and distinguish between different errors that may have occurred, and subsequently revert them. This process allows for a long lived quantum information preserving device: a quantum memory.

There exist different approaches to error correction, each with inherent advantages and hurdles. Here we are interested in the surface code, a specific type of error correction devices where the protection granted against errors comes from the size of the system we are using [1]. A key concept in the surface code, which we will repeatedly use to assess the effectiveness of the memory, is the concept of error threshold. The threshold marks the point after which increasing the resources used in quantum error correction will introduce more errors than it helps to protect against. That is, if the noise rate is greater than this threshold then error correction is no longer helpful.

When discussing the construction of a quantum memory it is essential that we also consider the physical medium through which it is brought into existence. Although actually there exist many promising candidates, here we will focus in Nitrogen vacancy (NV) centers in diamonds as the basic unit for our quantum memory construction. The main advantages of NV centers are the inherently long times quantum systems can be preserved in them [2] and the flexibility they have when transmitting quantum information between distant NV centers through the use of simple optical systems [3]. This makes a quantum network erected from NV centers the best candidate for a distributed approach to quantum computing. In this approach we consider a network composed of identical nodes connected to create a fundamentally scalable quantum memory. This scalability provides the perfect mechanism to construct a surface code, where increasing the

size of the system becomes a task of replicating the individual units of our network. Previous works show how this distributed approach can be done [4, 5], here we will expand on this by modeling a network consisting of NV centers. Where our first goal is to assess how capable is this NV formulated memory, and secondly to try different network constructions in order to improve the quantum memory.

Our analysis will consist first in a theoretical model of NV centers as quantum computers, where we will focus in the main error sources inherent to the physical system. This model will allow us to characterize each one of the error sources with a specific parameter, which we can vary to control the rate of noise introduced by that specific phenomena. Thereafter, we will study the operations and resources required to create the network in a manner required by the surface code. With this in mind we will use numerical simulations to drive our study of the distributed surface code under our NV center models. To begin with, we compare different methods in which the network can be realized, and simulate each one under different conditions in order to select the most effective one. With this results we continue to assess the effectiveness of the distributed implementation of the surface code through the values of the thresholds of the most relevant error sources. Lastly, we will devise new architectures by considering a network which utilizes more resources, namely more NV centers per node. The new architectures aim to improve the effectiveness of the surface code by reducing the noise induced into the memory.

We are motivated by the need to provide some understanding in how a quantum memory will fare under more realistic models of NV centers, in hope to provide insights on current efforts towards the development of NV centers as quantum computing devices. We achieve this by comparing the values of the error rates of our calculated thresholds with the ones measured experimentally. The results show that a initial implementation of the distributed surface code is still beyond the reach of a practical implementation, since the experimental values exceed considerably the threshold values obtained for all error sources. Nonetheless, we show that our new architectures allow for a more effective implementation of the surface code. For one architecture we show that by specially employing an additional NV center per node we can practically nullify the noise induced by one of the main sources. Furthermore, two architectures with 7 and 9 NV centers per node respectively show increasing improvements in the thresholds for the parameters of the remaining sources. Where some of the new found values are within range of the experimentally measured ones. These results provide key insights towards possible research directions to further develop a practical distributed quantum computer.

This thesis is organized as follows: in chapter 2 we give an overview of the theoretical concepts of quantum computing and quantum error correction. Specifically in the surface code, in the principles of its functionality and how can we simulate it to calculate the thresholds. In chapter 3 we will see the principles behind the physical implementation of the distributed memory. We

examine the use of NV centers as the unit for quantum computing and which are the noise sources that must be accounted for. We introduce the concept of a quantum network with the resources required and the methods through which it can be realized. From this we construct the model we use in the rest of the analysis. Chapter 4 contains all the concepts behind creating a distributed surface code, where we pay special attention to the details involved, and we present the results of our initial implementation of the distributed memory. In chapter 5 we formulate the new network schemes and we calculate the thresholds for each one of the new proposed architectures. We conclude by comparing the results and assessing the impact of these changes in the resistance of the memory to the different noise sources.

Fault tolerant quantum computing

To construct a practical quantum computer a method is required to protect and manipulate any stored information under the presence of errors. For this we must understand how to encode information on a error correction system in which we can deal with the errors without incurring in any serious propagation. Moreover, we must figure how to safely perform error correction when the operations used are themselves noisy. A protocol which performs these tasks is called fault tolerant.

In this chapter we will analyze the theory behind constructing a fault tolerant quantum memory. We begin by introducing the most basic fundamentals of quantum information theory, for a complete exposition we refer the reader to [6]. Subsequently, we discuss the concept of quantum error correction, and one of the main focus of this work: the surface code. Our analysis makes emphasis in each part involved in constructing a quantum memory, and the numerical methods employed to analyze the performance of this memory.

2.1 Concepts of Quantum Computing

Quantum computing is based on the principle of using quantum 2 level systems as quantum bits (qubits) as the elementary units for computing. In a quantum computer the computing is done over the quantum states of these qubits. Mathematically, a qubit state corresponds to a 2-dimensional complex vector in the Hilbert space \mathcal{H} [7] which can be visualized as a point in the surface of a sphere called the Bloch sphere [8]. Commonly the Z axis is used as the basis to express the states, as in $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, being $|0\rangle$ and $|1\rangle$ the vectors pointing in opposite directions along the Z axis, and α, β complex numbers. One of the essential postulates of quantum mechanics states that when measured any quantum state collapses into an eigenstate of the measurement operator. In the case of qubits, measurements are made along a certain direction with the eigenvalues 1 or -1 , for example when measuring along the Z the probability of obtaining the state $|0\rangle$ or $|1\rangle$ are $|\alpha|^2$ and $|\beta|^2$ correspondingly, it follows that in order to correctly represent the state $|\psi\rangle$ then is required that $|\alpha|^2 + |\beta|^2 = 1$.

Essential operations on qubits are expressed as unitary matrices, every operation can be portrayed as a rotation in the Hilbert space. The most fundamental operations are the Pauli matrices which correspond to π rotations along their respective X, Y or Z axis.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Similarly a state composed of multiple qubits corresponds to a vector in the n -dimensional Hilbert space \mathcal{H}^n , with n the number of qubits. In general a multiple qubit state cannot be expressed as a tensor product of single-qubit states, in this case we say the qubits are entangled to each other. Entanglement is a remarkable property which can describe physical phenomena that are inexplicable by classical mechanics [9]. The most common entangled states are the two-qubit Bell states [10]:

$$\begin{aligned} |\phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), & |\phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \\ |\psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), & |\psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \end{aligned}$$

The set of Bell states forms a basis that spans the two-qubit Hilbert space. These states can be created by means of what is commonly called the controlled gates, the most common are the CNOT and CPHASE gates.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{CPHASE} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

With one of these two-qubit operations, in combination with a set of single-qubit operations one can form a universal set of quantum gates in which any unitary operation can be decomposed [11].

2.1.1 Density matrices

Due to the nature of quantum mechanics, in practice it is almost impossible to have complete certainty at all times about a quantum state. Mathematically we require a formalism that allows us to deal with this uncertainty by being able to operate over statistical ensembles of quantum states. As such, we use the density matrix formalism in order to express a probabilistic

uncertainty of quantum objects $|\psi_i\rangle$. A density matrix ρ is a Hermitian matrix element of the operators of the Hilbert space $\mathcal{L}(\mathcal{H})$,

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

Density matrices satisfy the following properties: trace preserving $\text{Tr}(\rho) = 1$, that is $\sum p_i = 1$, and is positive semidefinite, which means that $p_i \geq 0$.

States that can be expressed by a single coefficient $p_1 = 1$, $\rho = |\psi\rangle\langle\psi|$ are called pure states. The rest of the cases are called mixed states, for example the n qubit maximally mixed state $\rho = \frac{1}{d}\mathbb{I}$ serves as the analogous of a classical state [12].

2.1.2 Channels

To introduce dynamics into density matrices we need to define the properties of general quantum operations, commonly called quantum channels [13], after the fact that they can be seen as operations that conduct information. Mathematically a channel is a linear map that transforms $\mathcal{N} : \mathcal{L}(\mathcal{H}_A) \rightarrow \mathcal{L}(\mathcal{H}_B)$, where \mathcal{H}_A and \mathcal{H}_B are different Hilbert spaces and $\mathcal{L}(\mathcal{H})$ denotes the set of linear operators on a given Hilbert space. For a quantum channel to be considered valid it must satisfy the following properties:

1. Is a linear operation, as required by the quantum mechanics formalism.
2. Is complete positive, that is if we introduce an n ancillary subsystem \mathcal{H}_n , the map $(\mathcal{N} \otimes \mathbb{I}_n)$ is also positive. Formally, $(\mathcal{N} \otimes \mathbb{I}_n)\rho \geq 0 \quad \forall \rho \in \mathcal{L}(\mathcal{H} \otimes \mathcal{H}_n)$, being \mathbb{I}_n the identity on the ancilla space \mathcal{H}_n .
3. Is trace preserving $\text{Tr}(\mathcal{N}(\rho)) = \text{Tr}(\rho)$.

2.1.3 Circuits

Quantum circuits are a very useful visual representation of sequences of quantum operations on multiple qubits. Figure 2.1 shows an example of a circuit that achieves quantum teleportation [14]. We can relate all mathematical objects discussed so far to the elements in a quantum circuit. The circuit begins with the initialized quantum states. The wires in each circuit represent the channel which transmits the state, where transmission is not limited to spatial transition, temporal transmission of a state, like a quantum memory, is also considered. The blocks or parts of a circuit represent operations acting on the qubit(s) specified by the channel(s) containing

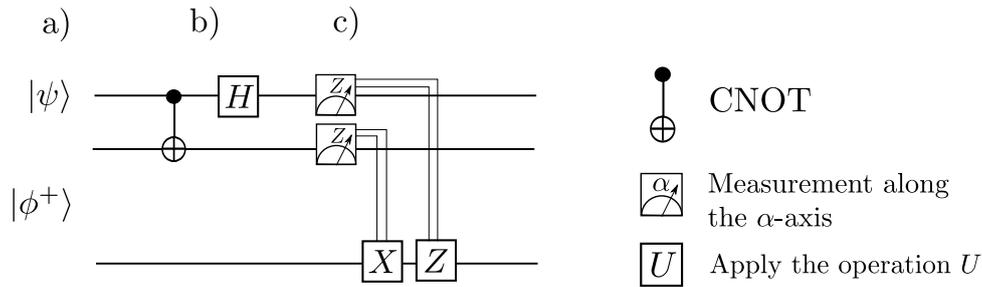


Fig. 2.1.: Circuits as a representation of quantum operations, the main parts of a circuit are: a) At the start of the circuit the quantum states are initialized in their respective channels. b) Unitary operations are performed on different qubits, the wire in which the operation sits signifies the qubit in which it applies. c) Measurements are made in the circuit in the basis indicated. The pipes connecting to other operations represent that this operations are turned on/off by the outcomes of the measurements.

the operation. When measurements of qubits are made, there is an implicit bit output and the collapse of the state in to the corresponding state according to the measurement outcome.

2.2 Quantum error correction

In order to deal with the unavoidable presence of noise in a quantum computer a method to perform error correction is needed. This method requires the ability to distinguish different errors in order to devise a operation that successfully amends the error. This is accomplished by using several qubits and operations to construct a bigger system capable of error correction, a process commonly called encoding. In mathematical terms error correction can be thought of as exploiting the redundancies of a larger Hilbert space where through non-destructive measurements we identify if unwanted operations on the physical system have occurred.

Although quantum error correction encompasses a larger set of methods, for simplicity we will restrict ourselves to the ones that can be described by the stabilizer formalism [15]. In this formalism we define a error correcting code through a set of operations corresponding to the error detecting measurements. These operations inherently also define what is known as the logical qubits of the code, the noise resilient qubits in the higher dimensional Hilbert space. Essentially, a code consists of n physical qubits used to encode k logical qubits in a subspace, called the code space, spanned by states $|\psi\rangle$ that are invariant under the action of the stabilizer group \mathcal{S} ,

$$P|\psi\rangle = |\psi\rangle \quad \forall P \in \mathcal{S}.$$

Generally the elements of \mathcal{S} are expressed as a subset of the Pauli basis \mathcal{P}_n , a group generated by the combinations of all Pauli matrices of n qubits including \mathbb{I} , and are commonly referred simply as stabilizers. Additionally, the stabilizers also define a set of logical operations that act in the logical states, as follows:

$$\begin{aligned} X_L |0\rangle_L &= |1\rangle_L, \\ Z_L |+\rangle_L &= |-\rangle_L. \end{aligned}$$

The distance of a code d is the minimum weight, number of single-qubit operations different to \mathbb{I} , of a logical operation. The numbers $[[n, k, d]]$ are commonly used in the literature to characterize a code. To exemplify how an specific error code is defined through the stabilizer formalism we can see the simplest of all error correction codes.

The bit flip code: When protecting against single-qubit X errors the data qubits can be encoded, through the application of two-qubit gates, in the logical basis $\alpha |0\rangle_L + \beta |1\rangle_L = \alpha |000\rangle + \beta |111\rangle$. In this code the stabilizer group is composed by the operators $\{\mathbb{I}ZZ, ZZI\}$, (note this is not a unique set and other ones could have been selected, for example $\{\mathbb{I}ZZ, ZI Z\}$). By using two-qubit gates and additional qubits (called ancilla qubits) we can make indirect measurements to obtain the eigenvalues of these operators. These measurements yield the information required to revert to code to the code space. That is, using the information extracted one can non deterministically identify if a single bit flip error has occurred and on which qubit, and thus revert the operation to correct the error. This is the core process of stabilizer error correction, commonly the act of obtaining the eigenvalues of stabilizer operators is referred to as measuring the stabilizers, and the outcome of the stabilizer measurements is called a error syndrome. We can observe that this code does not protect against a two bit flip errors, ex XXI which in fact corresponds to a logical bit flip X_L . Furthermore, a logical phase flip Z_L corresponds to any Z operations with weight equal to one, ex $III Z$. Then $d = 1$ for this code, then the only error correcting capabilities gained are against single bit flips.

A code must be able to deal with all types of error in order to be useful. The first code capable of this was introduced by Peter Shor in his famous 9 qubit code [16]. Later a smaller 7 qubit version [17] was presented. However, these codes fail when more than a single-qubit error occurs. Thus, we require a more sophisticated code to correct against errors on multiple qubits, in other words a code with a large distance.

2.3 Surface code

Introduced by Kitaev et al. in [1], as a way to topologically perform error correction in 2D. The surface code has risen to become the most famous error correcting code due to its high resistance to errors and the practical advantages of its physical construction. Being 2D means that it can basically be realized on a flat chip and thus current technology in the process of chip fabrication can be used. In general when speaking of the surface code there are two versions, the toric code which considers periodic boundaries, and the planar code with finite bounds. Previous works show how as distances become large the differences between them disappear [18]. Through this work we will mainly focus on the toric code.

More than a single error code the surface code represents a family of error correcting codes, each one categorized by the distance d . The surface code consists in a flat array of qubits as shown in fig. 2.2. The code is defined by the stabilizers composed from what is commonly called stars and plaquettes $\{\mathcal{S}_f^\star, \mathcal{S}_v^\blacksquare\}$, each one defined as:

$$\begin{aligned}\mathcal{S}_f^\star &= \prod_{i \in Q(v)} X_i, \\ \mathcal{S}_v^\blacksquare &= \prod_{i \in Q(f)} Z_i.\end{aligned}$$

Where $Q(v)$ represents the neighboring qubits of the vertex v in the lattice, and $Q(f)$ the qubits neighboring the face f . Under these stabilizers the logical qubits correspond to closed loops of operators connecting one end of the surface to another.

2.3.1 Faulty measurements syndrome extraction

We aim to restore the code space by measuring the stabilizers, obtaining a syndrome and performing some correction operation. When perfect measurements are available, the syndrome extracted by a single complete round of stabilizers allows us to find such correction. However, if the syndrome cannot be determined exactly because of the presence of measurement errors, correcting the code gets more complicated and more classical processing is required. For measurement errors we mean the possibility that the bit obtained after a measurement does not match the subspace in which the system collapsed. This normally happens when Pauli errors are present in the ancilla qubits.

Previous works [18, 19] have addressed this problem, where now in order to obtain a reliable syndrome a single measurement is not enough, but now several consecutive measurements must be made recording on each step the results obtained. This effectively adds a time dimension to

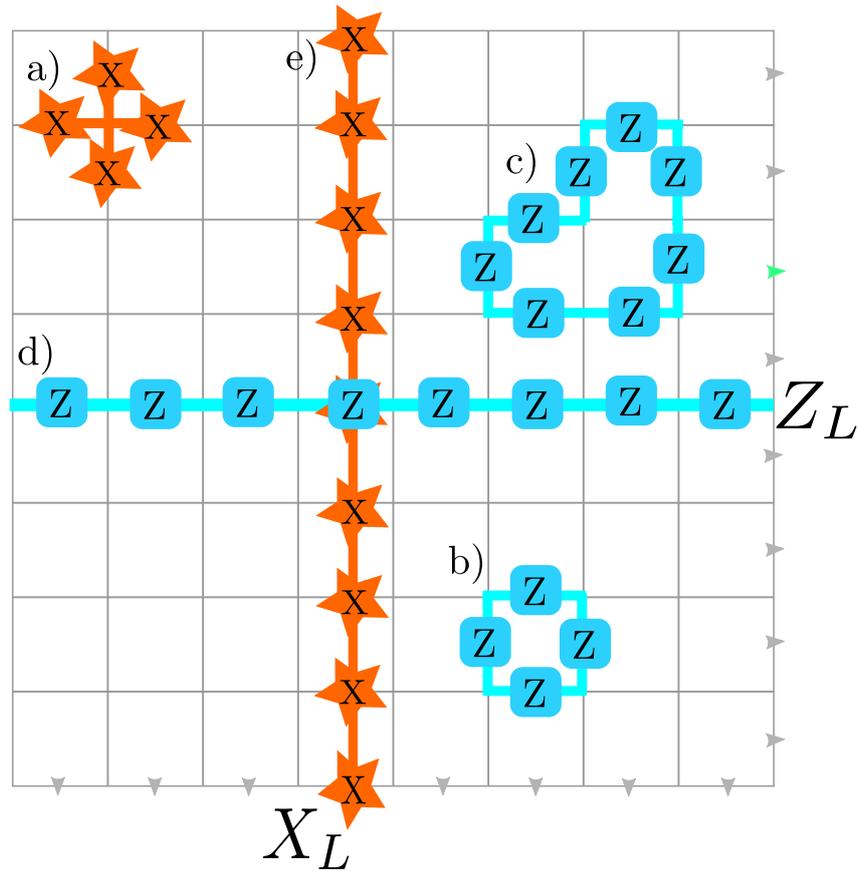


Fig. 2.2.: The surface code consists in a 2D sheet of qubits where data qubits are on every edge of the square lattice. In the toric code periodic boundaries are considered, in the planar code the boundaries are hard. Star stabilizers (a) are composed of X parity measurements and are located in the vertexes of the lattice. Plaquettes stabilizers (b) consist in Z parity measurements are located in the on the face of every square. The multiplication of two distinct stabilizers is also a stabilizer of the code (c). Logical operations Z_L (d) and X_L (e) connect the 2d surface in a non reducible way.

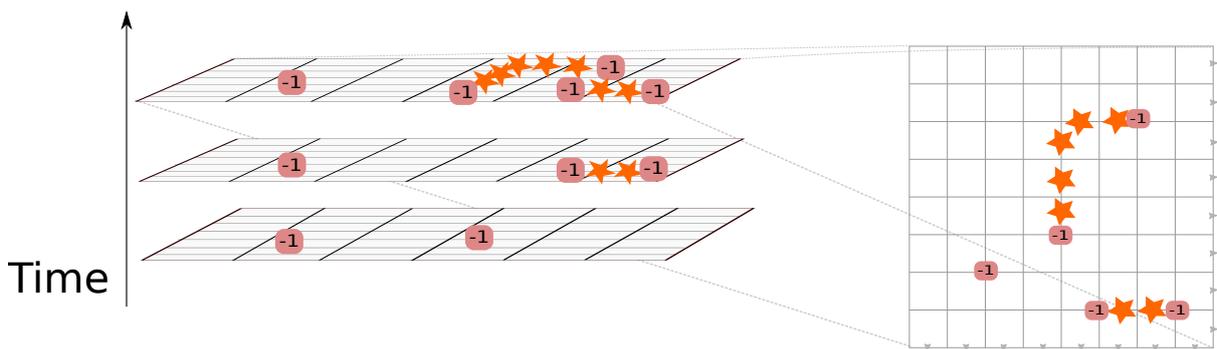


Fig. 2.3.: Measurements of the syndrome are obtained over time (left), each measurement is recorded in a different sheet of the surface code, adding a new time dimension to the coordinates of the measurements. Errors produce chains (right) in which only the ends produce a different stabilizer outcome.

the syndrome obtained, being each imperfect measurement a unit of time. Figure 2.3 shows a diagram of this process. The result obtained is a 3D array of measurements which must be decoded in order to find a correction operator that restores the code to the code space. In this 3D array, errors in the data qubits will create chains connected in the spacelike direction, in a single sheet of measurements at a given time, while measurement errors will create chains connected in the timelike direction. As we will see next, this distinction between chains due to the nature of the errors is what allows the decoder to successfully solve the syndrome.

2.3.2 Syndrome decoding

Now that the information has been extracted, a decoder is required in order to find the operations that return the code to the code space. We say that the correction has failed when there is a logical operation acting on the system after the correction has been applied. Several decoding algorithms exist for the surface code. Most notably, minimum weight perfect matching (MWPM) [20] and maximum likelihood decoding (MLD) [21]. Although MLD returns a higher success rate it is quite expensive numerically, especially when considering codes with larger distances. On the other hand, MWPM presents a good trade-off between numerical complexity and decoding success. Through this work we will use the MWPM algorithm.

In graph theory a perfect matching of a graph G is the subset of edges such that every node of G is met exactly by one edge. The MWPM algorithm is used to solve the problem of: given a graph in which all edges have a specific weight, to find the perfect matching of said graph with the subset of edges corresponding to the minimum weight. Then we see that the first step in using the MWPM algorithm is to translate our problem into a corresponding graph.

We extract the nodes marking the end points of every chain or errors from the syndrome recorded as the nodes of the graph. In figure 2.4(left) we can see a diagram of the resulting syndrome after several imperfect measurements have been made, with nodes added in the respective positions. Note that a node appears only when that point has changed from the immediate previous measurement. This is a crucial part as it not only characterizes the measurement errors, but allows us to make a distinction between chains of physical errors occurring in different slices of time, thus allowing us to correctly address every one independently. In order to correct for measurement errors we need to include virtual nodes in the time like direction, following the method used in [19], to allow the decoder to match errors in the time border of the syndrome. Thus for each syndrome point a extra virtual point must be created on the outside of the time border. In the special case of the planar code, virtual nodes must also be added in the borders of the plane to account for the chains ending on the border.

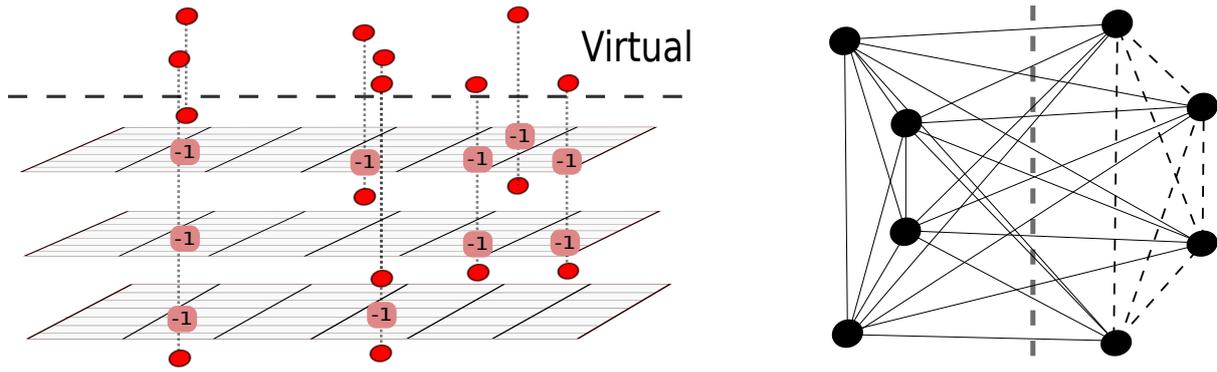


Fig. 2.4.: Nodes represented by red dots are produced from the syndrome measurements (left), a node is placed after every stabilizer measurement changes from value 1 to -1 or vice versa. Virtual nodes are placed over every node with the time coordinate displaced to the $t + 1$ sheet of measurements. (right) With the nodes a fully connected graph is created where the values of the edges correspond to the Manhattan distance, with the exception of the edges connecting two virtual nodes which have value 0 represented by a dashed line.

To obtain the set of edges with their respective weights we use the Manhattan distance between each node to weight each edge. The Manhattan distance is defined as the distance between two points following strictly a square grid to connect them, it is the simple sum of the horizontal and vertical components of the shortest path between the points along the grid. In order for the MWPM algorithm to work we make the weight of every edge between two virtual nodes to be equal to zero, thus making the matching between two virtual nodes “free”. Figure 2.4(right) shows an example of the graph obtained from a syndrome.

Since we are dealing with imperfect measurements and the number of measurements performed is finite, there is always some uncertainty in the types of errors detected over the last measurements. In these the decoder is unable to determine if the last errors are produced by physical errors or by measurement errors. To deal with this, in [1] the authors propose a procedure called *overlapping memory recovery*. The main idea is not to over react to syndrome information that is potentially faulty, but to take action only in the long lived errors seen by many measurements and to keep performing measurements and decoding as time goes on. This is however when considering a more realistic surface code, in our simulations, since our final goal is to evaluate if the decoder introduced a logical error in the correction operator C , we can dispel this uncertainty by making use of a round of perfect measurement as explained in the following section.

2.3.3 Thresholds

The thresholds are a quantity in the error rate that allows us to characterize every error correcting code. In general as defined by [15] we speak about a pseudo threshold $p_{ps}(d)$ as the point in

which below it the logical error rate of the code is smaller than the qubit error rate, and vice versa above. In other words, if the error rate is below the pseudo threshold correcting helps, otherwise it increases the occurrence of logical errors. When talking about the surface code since we are addressing a family of error correcting codes, we will talk about the threshold p_{th} , which marks the limit point after which increasing the code distance increases the logical error rate. That is, if the error rate is below p_{th} then we can increase the distance of the code to reduce the logical error rate as much as possible. It is easy to see that indeed when $d \rightarrow \infty$ then $p_{ps}(d) \rightarrow p_{th}$.

Following the results from [22], the behavior of the logical error rate near the threshold can be related to that of the critical parameters in a physical system known as the spin model when it is near phase transition [23]. Then making an analogy between the spin correlation length near phase transition and the logical error rate p_l near the threshold we know that:

$$p_l \approx |p - p_{th}|^{-\nu_0}.$$

Being p the physical error rate and p_{th} the threshold in the thermodynamic limit (the limit of infinite distance) and ν_0 a scaling exponent. As such for sufficiently large code distance, the logical error rate should follow the same behavior

$$p_l = (p - p_{th})d^{-\nu_0}.$$

In order to allow for finite size errors we fit the data to the function

$$p_l = A + B(p - p_{th})d^{1/\nu_0} + C(p - p_{th})^2 d^{2/\nu_0},$$

to find the value of p_{th} .

2.3.4 Numerical simulations

In order to find the thresholds we perform Monte Carlo simulations following the methodology described in [5] to simulate a surface code in which errors occur at a certain rate. We make use of Kolmogorov's Blossom V algorithm [24] to solve MWPM in the decoder. The code used during this thesis works on the basis of the stabilizer formalism and can be found documented in [25]

As mentioned before, since we deal with imperfect syndrome extraction, if we were to use the method of overlapping memory recovery the process of measuring and decoding would be a continuous one. Here due to the nature of the simulations we are restricted to a finite number of imperfect measurements. Thus, in order to obtain a definitive result we must make use of a final round of perfect measurements. This allows us to determine if a logical error has occurred after the decoding. A physical justification of the perfect measurement is that after the last stabilizer

measurement, we can additionally measure all data qubits of the surface code and compare the results with the syndrome, effectively yielding perfect syndrome information.

The steps followed to simulate the surface code are:

1. Initialize the system with all the qubits on the +1 eigenstate of all stabilizers.
2. Perform N rounds of measurements and record the resulting syndrome, while introducing randomly physical and measurement errors in between each round .
3. Perform a last perfect measurement to obtain the complete syndrome, as shown in fig. 2.5 (left).
4. Decode and apply the corrections to the surface code, as in fig. 2.5 (right).
5. Evaluate the data qubits to find if a logical gate has occurred on the code.

The result after this steps is fail if an logical operation happened and success otherwise. In order to fully approximate the value of the logical error rate p_l several iterations of this procedure must be realized. Since we are dealing with a Bernoulli distribution we can make use of a special case of Hoeffding's inequality [26]. In which if we perform a total of N times this procedure and if $F(N)$ is the number of times the procedure returns fail then,

$$\Pr(F(N) \leq (p_l - \epsilon)N \text{ or } F(N) \geq (p_l + \epsilon)N) \leq 2 \exp(-2\epsilon^2 N).$$

That is, in order to have an accuracy of $\epsilon = 1\%$ and a precision $< .5\%$ we require $N = 3 \times 10^4$ iterations.

To test our simulations we simulate a simplified isotropic error model in which independent spin and phase flips appear on the physical qubits with an equal rate as measurement errors. To find the threshold we will use the logical error p_l rate per d measurement rounds, following the methodology in [22]. The result as seen in figure 2.6 return the threshold values:

$$\begin{aligned} \text{toric: } p^{th} &= 0.0292 \pm 0.0002, \\ \text{planar: } p^{th} &= 0.0297 \pm 0.0002. \end{aligned}$$

Our results for the toric code coincide with those obtained in [22]. The planar code shows a higher threshold value, which can be understood as explained by Wang et al. [18] as the finite dimension effects in which the border stabilizers, since they only interact with 3 qubits, are more protected against noise. Nonetheless, if we were to increase the distance of the codes, in our

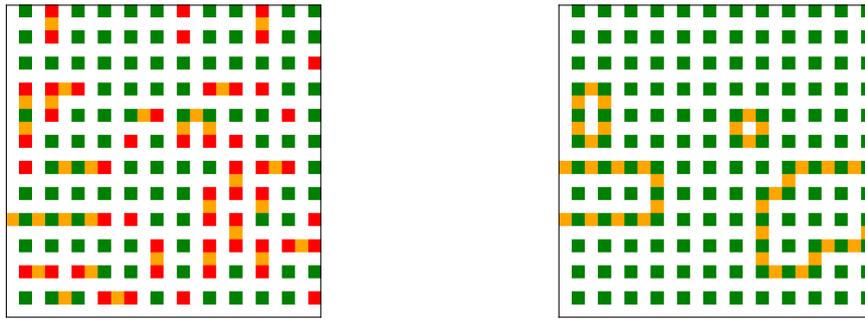


Fig. 2.5.: (right) The simulation randomly produces errors on the data qubits marked in orange and measures the stabilizers which produce a syndrome. Measurement values 1 are marked in green while -1 values are marked in red. Errors in the measurements may produce stabilizer false positive error values where there are no physical errors. (right) After the syndrome is decoded correcting operations are performed to bring back the surface code to the code space.

numerical simulations we would expect to see the threshold of the planar code converging to the one achieved by the toric code.

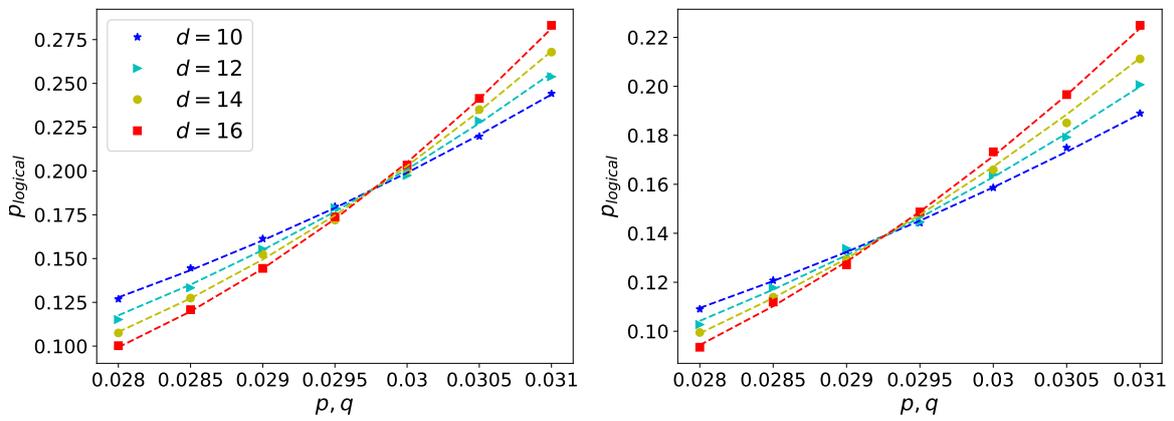


Fig. 2.6.: Thresholds for the surface code under random errors and imperfect measurements. The physical error rate p is taken equal to the measurement error rate q . (left) Threshold of the planar code. (right) The threshold for the toric code.

NV centers as quantum nodes

In order to construct a quantum computer we need to consider a physical system as the hardware basis in which the qubits are realized. To this day there exist many possible physical systems for this purpose, the most prominent ones are superconducting qubits [27], ion qubits [28] and nitrogen vacancy (NV) centers in diamonds [29]. Throughout this work we will focus on NV centers. The motivation derives from their large decoherence times and possibilities for scalability [3, 2]. This makes NV centers the best candidate when constructing a distributed quantum memory.

In this chapter we will overview the basic physical phenomena exploited to allow for quantum computing in NV centers. Including the physical limitations and noise sources imposed by the hardware. With this in mind, we motivate an error model which allows us to construct a numerical model, while preserving a sensible representation. Following we will see how a quantum network based on NV centers can be constructed. Making special emphasis into the methods used to generate entanglement through the network.

3.1 NV centers in quantum computing

A Nitrogen Vacancy (NV) center is a lattice defect in a diamond crystal lattice, in which a nitrogen atom and a vacancy substitute two neighboring lattice sites in the diamond. This results in five unbound valence electrons, with two originated by the nitrogen atom and three from the carbon atoms. Additionally an extra electron can be captured from the environment, forming a negatively charged NV center state. This six electrons occupy the molecular levels forming a spin-triplet ($S = 1$) through the spin-spin interactions. This triplet through the control of an external magnetic field splits the energy levels of the $m_s = 0, \pm 1$ spin. Using this electron-spin qubits can be controlled using the levels $m_s = -1$ and $m_s = 0$. Additionally to this electron-spin, there is a natural abundance of $\approx 1\%$ of spin $1/2$ ^{13}C isotopes which can be used as nuclear qubits which are addressed by the electron-spin. The electron-spin qubit and nuclear qubits are mostly surrounded in the lattice by ^{12}C spin-less carbon atoms, which shield each individual qubit from each other and the environment. This shielding yields nuclear qubits with high coherence times which need to be addressed through the electron-spin qubit. For a more complete overview

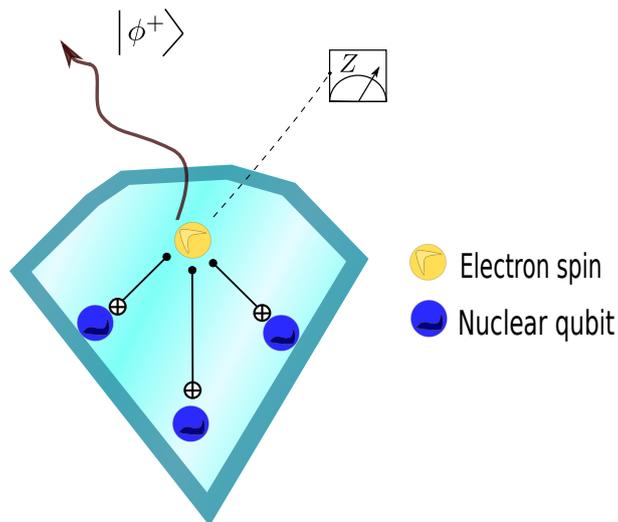


Fig. 3.1.: NV center as the elemental device in quantum computing. Noise resilient nuclear-spin qubits are used as quantum memories. The electron-spin qubit is used as a communication device being able to remotely generate entanglement and perform state initialization, measurements and local two-qubit operations involving the nuclear qubits.

of the physical phenomena and properties of NV centers in quantum computing we refer the reader to [30].

With the physical system in mind our task now is to extract a model of NV centers as a quantum computing device which can be simulated to extract the required information.

3.1.1 Hardware limitations

Using NV centers as the unit of quantum computing we must account for the physical limitations these impose when carrying operations, fig. 3.1 exemplifies the qubits and which operations can be carried on them in a NV Center. The intrinsic limitations that we must take into consideration are:

- Only the electron-spin qubit can be used to generate entanglement.
- Two-qubit operations can only be made between electron-spin and another nuclear-spin qubit
- Measurements can only be made in the Z basis on the electron-spin qubit.

To overcome the fact that almost every operation inside the node has to be done through the electron-spin, SWAP operations are commonly required between the electron-spin qubit and the nuclear qubits. This SWAP operation is normally carried by using three CNOT gates, see figure 3.5

3.2 Error models

We need to include error models in order to account for the different sources of noise in our quantum system. A NV center as a quantum node is a very complex system where if one accounts for all the required hardware and operations there are countless sources of noise inducing phenomena, each with their corresponding parameters. While in theory is possible to consider each one, in most cases several phenomena can be grouped together into a single noise effect from which a general error model can be considered. In this project we constrain ourselves to a simplified model which includes the most relevant sources of noise but that is complete enough to allow us to draw conclusion about a practical implementation using NV centers. The error sources we consider are:

- **Single-qubit gate:** Every noisy single-qubit gate is treated as a perfect gate followed by depolarizing noise, such that with probability p_s one of the three Pauli operations applied to the qubit.

$$N_{G1}(\rho) = (1 - p_s)\rho + \frac{p_s}{3} \sum_{i=1,2,3} \sigma_i \rho \sigma_i^\dagger. \quad (3.1)$$

- **Two-qubit gate:** The noise is modeled as a perfect gate followed by one of any of the Pauli operations on two qubits applied at random with probability p_g .

$$N_{G2}(\rho) = (1 - p_g)\rho + \frac{p_g}{15} \sum_{(i,j) \neq (0,0)} (\sigma_i \otimes \sigma_j) \rho (\sigma_i \otimes \sigma_j). \quad (3.2)$$

- **Environment error on nuclear-spin qubit:** To model the intrinsic noise induced to the qubits by its environment in the case of NV centers, we distinguish between two sources of environmental noise. One is the intrinsic noise induced by the environment to the qubits. The second one is a stronger depolarizing effect caused by the hyperfine interaction with the electron-spin that occurs when the electron-spin qubit has a non-zero angular momentum ($m_s = 1, -1$). This is the case when the electron-spin qubit is in a superposition

state, which occurs most prominently when entanglement is being generated between two distant NV centers. We model this two effects by means of the depolarizing channel:

$$D(\rho) = \frac{3\lambda + 1}{4}\rho + \frac{1 - \lambda}{4}(X\rho X + Y\rho Y + Z\rho Z), \quad (3.3)$$

with $\lambda = e^{-at}$, where the parameter a depends on the physical operation that is being carried at the moment by the node:

- If the NV center is being used to generate entanglement: $a = (a_0 + a_1)$
- Else, NV center is in standby, only memory error: $a = a_1$

With a_0 the parameter corresponding to the depolarizing introduced by the techniques used to generate entanglement and a_1 the parameter corresponding to the intrinsic memory error of NV centers. The complete channel of all the qubits is taken as a tensor product of the individual qubit depolarizing $D_{NV} = D^{\otimes n}$.

- **Measurement error:** For the measurements, with a given probability p_m called the measurement error rate, the outcome of the single-qubit measurement will be flipped. We consider measurement errors by replacing the projectors involved in a measurement by the imperfect projectors, which are expressed as:

$$\tilde{\Pi}_q = (1 - p_m) |q\rangle\langle q| + p_m |\bar{q}\rangle\langle \bar{q}|. \quad (3.4)$$

With q the desired state to be projected on and \bar{q} the orthogonal one.

3.3 NV centers in a network

We have described the physical principles of how NV centers can work as a elementary quantum computer. Now we need to describe how can this work as a node which connects to others to form a network in which a quantum memory can be realized. For this we consider first a scheme with two NV centers per node, one is designated to the task of creating and storing entanglement. While the other will hold the data qubits in which information is stored. This quantum memory can effectively benefit from being encoded in noise protected subspaces as in [2]. In chapter 5 we will relax this condition by considering additional NV centers per node to generate entanglement. Figure 3.2 shows a schematic of a network composed of NV centers. In addition to the nodes some degree of classical computing capabilities are required through the network to be able to perform error correction.

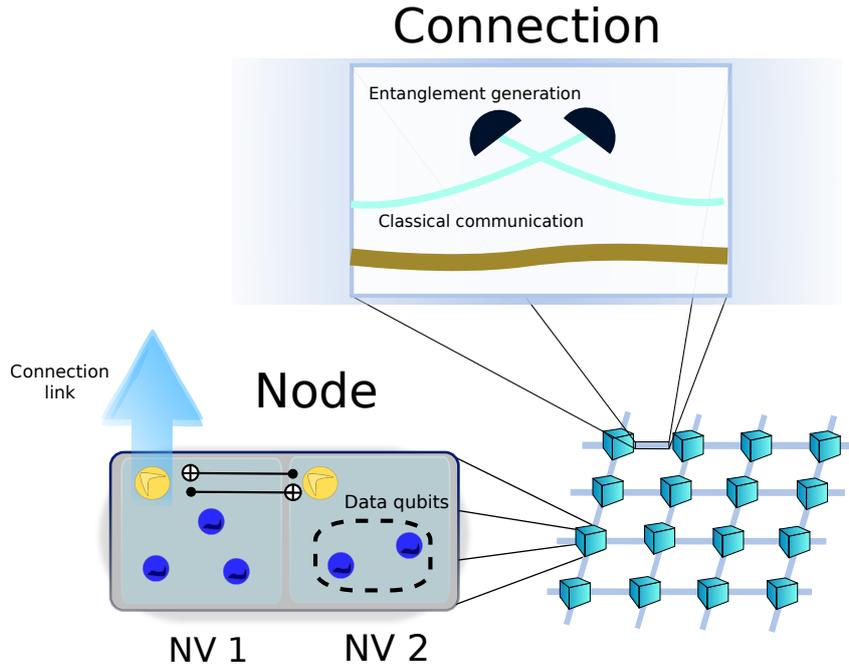


Fig. 3.2.: A NV center based quantum network composed by nodes connected to their nearest neighbors. Links consist in a classical communication channel, as well as a optical device capable of generating entanglement. Each node contains two NV centers, one used to generate and store entanglement while the other is used uniquely as a quantum memory.

The links of the network consist on a classical communication channel, as well as the optical elements required to create entanglement between the nodes. Figure 3.3 shows the main optical elements involved in creating entanglement. Each element is considered *lossy*, meaning there is a probability of loosing a photon as it transverses each element, consequently we can group the values as a single parameter which will be used in our model of the network.

$$\eta = p_{em} \cdot p_{ps} \cdot p_d \cdot \eta_f \quad (3.5)$$

Since we consider the nodes to be in cm or mm of distance apart from each other we can approximate the transitivity of the optical fiber to be $\eta_f = 1$. This simple characterization of the entire process by a single transitivity parameter is physically accurate as most of the phenomena involved in the entire process can be translated into photon loss.

3.3.1 Entanglement generation

With the network elements specified, we still need to address the question of how entanglement can be generated remotely between nodes. This is made by using a joint measurement between two photons originated from remote nodes that are brought to interact inside a beam splitter

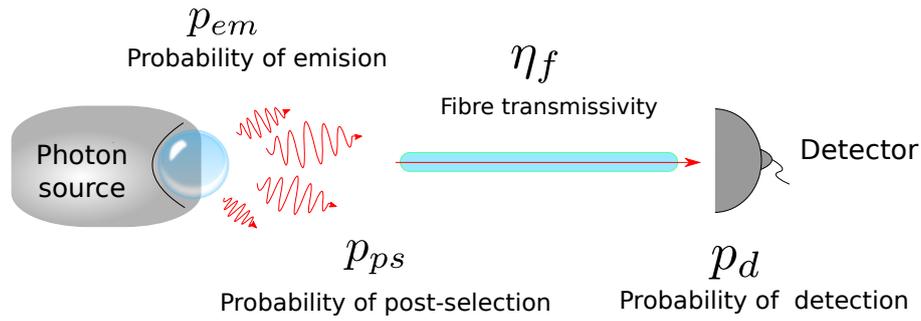


Fig. 3.3.: The most relevant parts involved in the optical device used to generate entanglement between nodes. Each part is modeled by a characteristic probability of successfully transmitting the photon to the next one. Starting from left to right: the probability of a photon source to emit a photon within the desired time bin to be used is given by p_{em} . Post-selection with success probability p_{ps} is made to the photons to ensure they are identical. The photons are transmitted through an optical fiber with transmissivity η_f to a detector with probability of detection p_d .

[31] this is the commonly called single photon protocol. Previous works show how can this method be realized in the laboratory using NV centers [3].

The single photon protocol is made between two distant nodes and requires one optically active qubit in each node. In the NV center context those qubits correspond to the electron-spin. This two qubits are first initialized in the state

$$|\psi\rangle = \cos \theta |0\rangle + \sin \theta |1\rangle ,$$

As shown in fig. 3.4 the $|1\rangle$ component of the state can be excited to produce a photon and then decay back. The emitted pair of photons are transmitted through a fiber into a beam splitter after which there are two detectors. When both photons reach the beam splitter they interfere with each other through a phenomena known as the Hong-Ou-Mandel effect [32]. Which describes the phenomena when two identical photons enter a 50:50 beam splitter. In short there are four possible scenarios for the photons to behave: 1) The photon coming in from above is reflected and the photon coming in from below is transmitted; 2) Both photons are transmitted; 3) Both photons are reflected; 4) The photon coming in from above is transmitted and the photon coming in from below is reflected. If both photons are identical in their physical properties and reach the beam splitter at the same time then only scenarios 1 and 4 should occur, namely both photons will exit towards the same direction chosen at random. This is the effect directly responsible for entangling the two matter qubits.

After the photons go through the beam splitter they are detected by two detectors, the information given by the detectors signals if the entanglement protocol was successful or not. For this reason this process is commonly referred to as heralded entanglement generation. In the case of no photons emitted or two photons detected the remaining qubits are left in un-entangled states

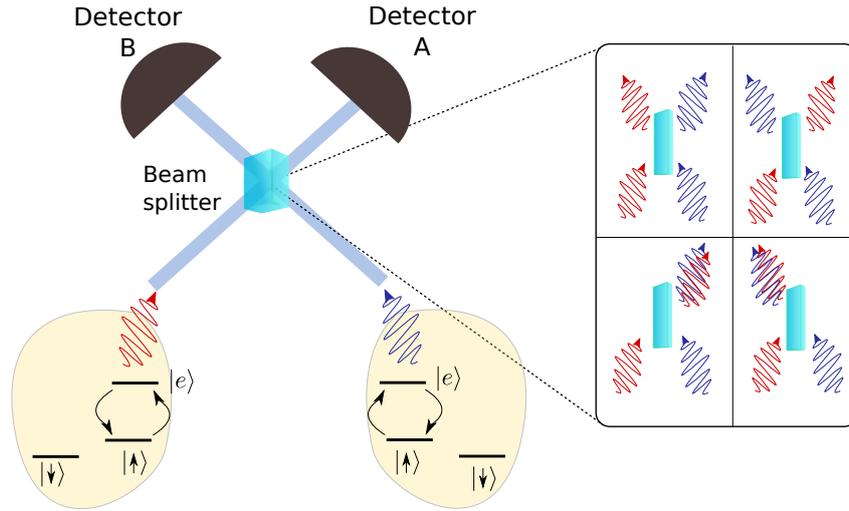


Fig. 3.4.: Entanglement generation via two optically active nodes (left). Each node contains a matter qubit in which the $|\uparrow\rangle$ is excited in order to emit a photon. The emitted photons interact through a beam splitter (right) in which there are four respective outcomes as explained in the text. After the photons interact in the beam splitter they are detected by two detectors at the end of the fibers.

$|00\rangle\langle 00|$ or $|11\rangle\langle 11|$. Nonetheless, in the case of only one detection the qubits yield the entangled state

$$|\Psi_{\theta}^{+}\rangle = \frac{1}{2} (|01\rangle + e^{i\theta} |10\rangle).$$

With the extra phase $e^{i\theta}$ due to the optical distance difference in the setup. This is however considering the process is ideal, in reality as discussed the entire process depends on the many parts of the optical setup. Under the small distance consideration and the lossy network model we denote the entanglement generation rate parameter η , and we must account for photon loss errors. This is the case when both photons do not exit towards the same direction, but one is lost before being detected, leaving the matter state qubits not entangled. This gives us a final state,

$$\rho_{raw} = (1 - r) |\psi^{+}\rangle \langle \psi^{+}| + r |11\rangle\langle 11|.$$

With $r = \frac{(1-\eta)\sin^2\theta}{1-\eta\sin^2\theta}$ and $|11\rangle\langle 11|$ corresponding to the un-entangled state of the matter qubits. The probability of success is thus:

$$P_{succes} = 2 \sin^2 \theta (\cos^2 \theta \eta + \sin^2 \theta (1 - \eta)).$$

And the fidelity of the state is given by $F = (1 - r)$. The short distances between nodes also allows us to ignore other error sources such as dark counts as shown in the analysis presented in the appendix of [4].

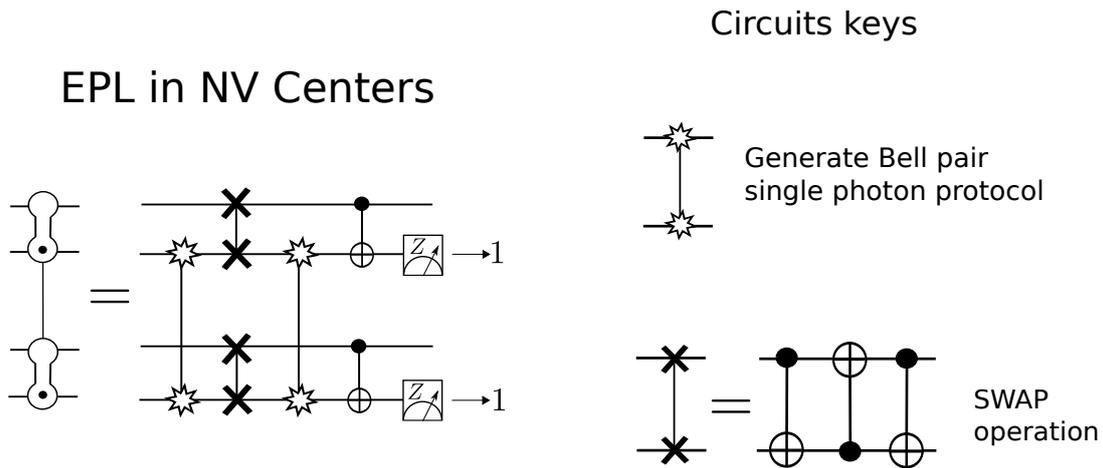


Fig. 3.5.: (right) Symbol to represent the circuit for performing the EPL protocol, the protocols is successful when both of the measurement result in 1. (left) Nomenclature used to represent the operations of entanglement generation via the single photon protocol and a SWAP operation.

3.3.2 Barret-Kok

The Barret-Kok (BK) protocol, consists in using two consecutive single photon events to guarantee the resulting state is entangled [33]. The BK scheme works by applying single-qubit rotations $X \otimes X$ on both matter qubits after the first event and then requiring a second single photon event. This protocol is extremely robust against the most important types of errors such as emission loss and detector loss. Thus, the result is a Bell state with fidelity is $F = 1$ under the small distance assumption. Here we require $\theta = \pi/4$ to maximize probability of success, which is given by:

$$P_{success} = (1 - r)\eta^2.$$

Because of the quadratic dependence in the detection probability the probability of success is considerably smaller that in the single photon protocol.

3.3.3 Extreme photon loss

The extreme photon loss (EPL) protocol deals with the error induced by photon loss by generating two entangled states and consuming one to verify the other one. Figure 3.5 shows the circuit that performs the EPL protocol. If perfect operations are considered the success rate for this protocol is $R = \frac{1}{2}(1 - r)^2 p_d$. However, in the more realistic scenario we are considering the result is also dependent on the error rate in the two-qubit gates and measurements, as such it must be calculated each time. Additionally, the depolarizing occurring to the first state due to the waiting time it takes to generate the second needs to be taken into account.

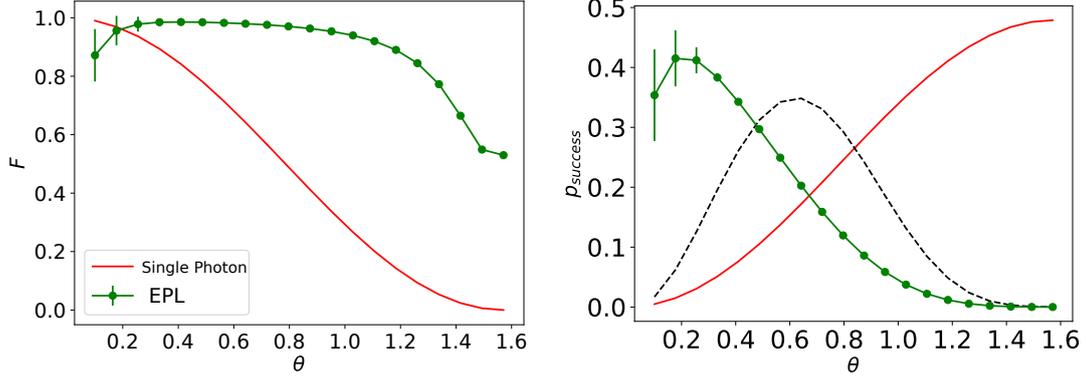


Fig. 3.6.: (left) Fidelities for the single photon method and the EPL protocol. For the EPL the results are the average over 300 simulations, in each case total time required for the generation of each pair is different. (right) Success probabilities for the two methods is shown, in the single photon method the curve is magnified by a factor of 100 for visualization purposes. For the EPL the probabilities shown represent only the probability of measuring 1 in both qubits. The probability of success of the entire protocol, the product of the two, is shown by the dotted line magnified by a value of 1000.

In figure 3.6 we compare the fidelities and probabilities of success and fidelities for the EPL protocol using the reference values shown in table 4.1 and varying the initial state parameter θ . We can see that the maximum success probability of the entire protocol is reached when $\theta_m \approx 0.62$. Most notably we observe that the behavior of the fidelity is almost flat near θ_m , and thus fixing $\theta = \theta_m$ will yield the best method for generating entanglement.

3.4 Parameters

To summarize, with the error models proposed above now we list all the error parameters required that will be fixed in the simulations, additionally we show some reference values to give some insights on the magnitudes of each one.

- Environmental error:
 1. Interaction noise electron-spin with nuclear qubits a_0 - *reference:* 83.3 [34].
 2. Memory error a_1 - *reference:* $\frac{1}{3}$ [34].
- Photon detection probability: η - *reference:* $\frac{1}{2000}$ [3].
- Entanglement generation attempt time: $t_{entanglement}$ - *reference:* $6\mu s$ [34].

- Operation time (two/one qubit gates, measurements): $t_{operation}$ - reference: $200\mu s$ [2].
- two-qubit gate: p_g - reference: 0.006 Thresholds values on [5].
- Measurement error rate: p_m - reference: 0.006 [5].
- single-qubit gate: p_s - reference: 0.006 [5].

3.5 Outline

Although the error models proposed above cover the main decoherence error effects on a quantum network composed of NV centers, there are still some secondary effects that we have not included in our analysis. This secondary effects arise from current experimental methods used in the operations required to control the qubits. It would be of interest in future work to further include this effects for a more complete analysis. One of these secondary effects occurs under the current methodology for performing measurements in the electron-spin. In which when a measurement error occurs there is additionally a probability p'_m in which the nuclear-spins suffer additionally a correlated error as shown by the channel:

$$D_{NV}(\rho) = (1 - p'_m)\rho + p'_m \sum_i a_i Z^i \rho Z^i, \quad (3.6)$$

The index i denoting the positions of the nuclear qubits and a_i the probability of a phase flip occurring in each nuclear qubit, with $\sum_i a_i = 1$.

Distributed surface code

So far we have seen how a noise resilient quantum memory can be achieved using the surface code, and we discussed how to model a quantum network constructed with NV centers. Now we analyze the methods followed in which we can construct a surface code implemented over a network composed of NV centers. To this end we will follow the example of previous works [4, 5] in which multipartite entangled states are used in the stabilizer measurements required by the surface code.

In this chapter we will first consider how this non-local measurement is achieved through entanglement and the different methods available to generate that entanglement. We continue analyzing the protocols used to generate multipartite states efficiently over the parts involved in each stabilizer measurement. We will explain the methodology, followed by numerical simulations to analyze the different protocols and present the results obtained of comparing the protocols. Lastly, we will present threshold calculations over different parameters to assess the effectiveness of the first implementation of the distributed surface code.

4.1 Distributed stabilizer measurements

In order to create the surface code in a network we require a method to perform non-local stabilizer measurements. To achieve this we follow the procedure presented in [5] where a GHZ distributed over the involved nodes is used. Without loss of generality, consider we want to perform the stabilizer measurement corresponding to a plaquette, or a Z basis parity measurement. We begin with a previously created shared ancilla GHZ state between four nodes and the state $|\psi\rangle$ of the data qubits,

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n} \right) |\psi\rangle.$$

Applying a local CPHASE gate between each one of the ancilla qubits and the data qubits we obtain:

$$\frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes n} |\psi\rangle + |1\rangle^{\otimes n} \bigotimes_{i=1}^n Z |\psi\rangle \right).$$

Where we used that $\text{CPHASE} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes Z$. Now transforming the ancilla qubits to the X basis gives us:

$$\frac{1}{\sqrt{2^{n+1}}} \left[\sum_{x \in H(x) \text{ even}} |x\rangle \left(\mathbb{I} + \bigotimes_{i=1}^n Z \right) |\psi\rangle + \sum_{x \in H(x) \text{ odd}} |x\rangle \left(\mathbb{I} - \bigotimes_{i=1}^n Z \right) |\psi\rangle \right].$$

With $H(x)$ the Hamming weight of the bit string x that represents every ket in the computational basis. We can recognize the effect of the gates over the state $|\psi\rangle$ is that of the projectors to even and odd parities in the Z basis,

$$\frac{1}{\sqrt{2^{n+1}}} \left[\sum_{x \in H(x) \text{ even}} |x\rangle |\psi_{\text{even}}\rangle + \sum_{x \in H(x) \text{ odd}} |x\rangle |\psi_{\text{odd}}\rangle \right].$$

After measuring in the X basis the ancillas we can see that the parity outcome of the state will be encoded in the Hamming parity of the resulting bit string and the state will have collapsed into a subspace of the corresponding parity. We must observe that the number n is not fixed nor required to have a specific value, we can see that the case $n = 1$, equals the monolithic case in which all gates implemented are local to the ancilla. This means that depending on the way the data qubits are distributed through the network we can find a way of performing the stabilizer measurements using GHZ of different weights. We will use this to use lower weight GHZ states in the stabilizer measurements on chapter 5.

For a schematic comparison, in figure 4.1 we can see how a stabilizer measurement can be performed in both the monolithic and distributed architectures. We can appreciate the increased resources required by the distributed approach, most notably the fact that a multipartite entangled state must be shared between all the parties involved in the stabilizer measurement.

Finally, here we considered a perfect GHZ state when doing the stabilizer measurement. We can see how in reality this will be a mixed state, which in combination with the imperfect operations will propagate errors to the data qubits and have a probability of resulting in an incorrect measurement. Correcting this errors is the main task of our surface code implementation. For our study this means that first we require to simulate the creation of the GHZ states from which we can model the noisy stabilizer measurement. This model will give us the probabilities of errors occurring in the surface code, from which we can simulate the surface code as specified in the previous chapter. The theoretical details of how we extract the exact probabilities for each error are presented in appendix B.

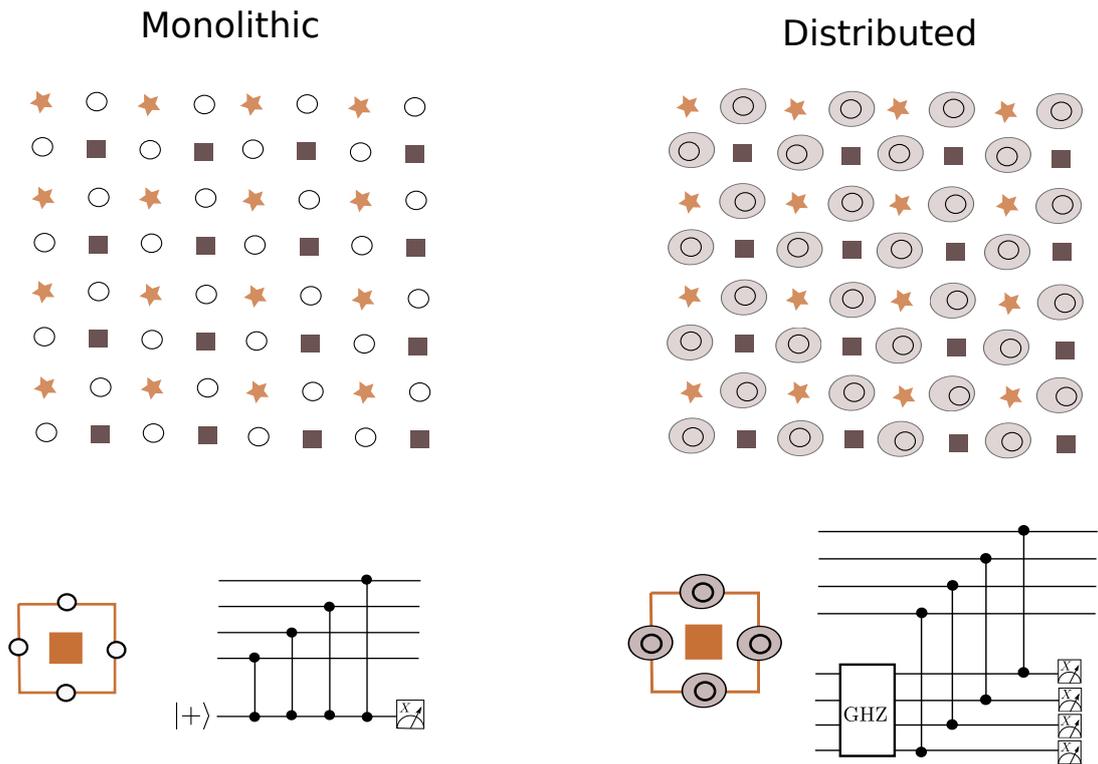


Fig. 4.1.: Example of a plaquette stabilizer being measured in the monolithic architecture (left), where all operations used are local. The distributed surface code (right) requires a fourpartite GHZ state to be produced between all the nodes (shaded area) before each stabilizer can be evaluated.

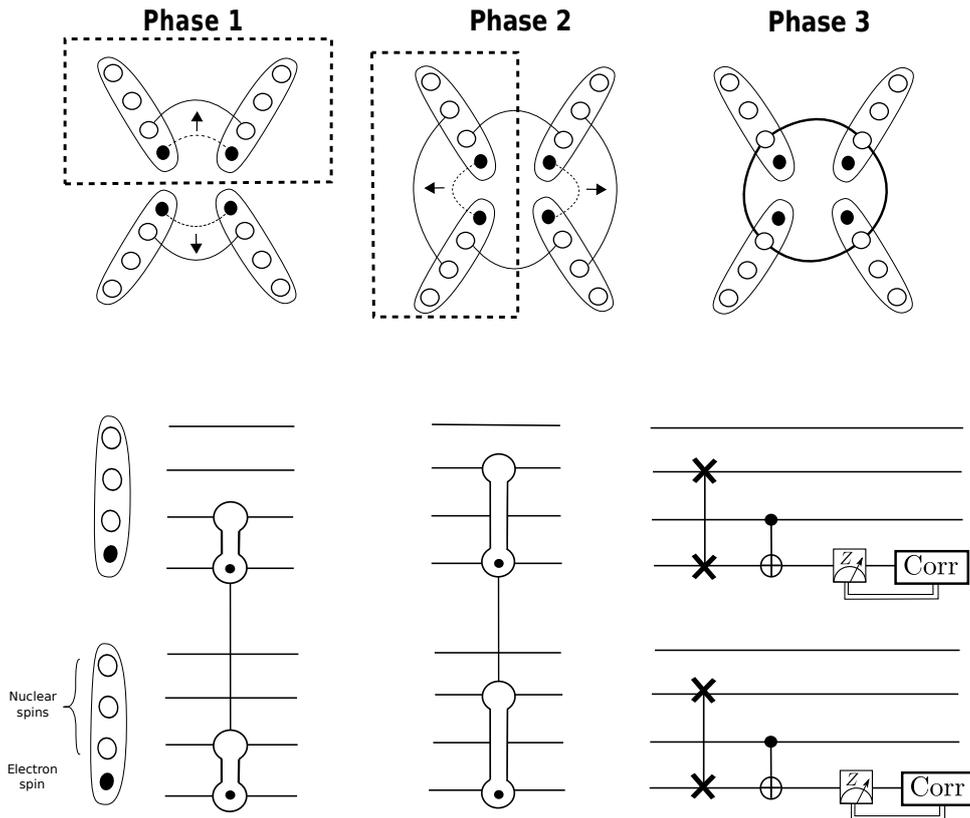


Fig. 4.2.: Protocol diagram for creating a weight four $|GHZ\rangle$ between four nodes. We use the notation for creating entangled pairs via the EPL protocol 3.5. In phase 1 two entangled pairs are generated in parallel and stored by the nodes with their left/right neighbor. In phase 2 another 2 pairs are generated but now with the top/bottom neighbor. In phase 3 local operations are made over the qubits involving the 4 Bell pairs and one qubit is measured over each node. Corrections may be applied depending on the result of this measurement.

4.2 GHZ state generation protocols

Now we outline the general method for generating a GHZ state between the nodes. Here we will only exemplify the case for a weight four GHZ state, although the procedure can be easily reduced to create GHZ states of weight three. As shown in figure 4.2, the process consists in two rounds of entanglement generation between the nodes, with corresponding SWAP gates in between the rounds. Followed by local two-qubit operations and measurements. In the last step depending on the measurement outcomes extra single-qubit rotations may be required to obtain a GHZ state. The details and the specific calculation for this process is shown in section 3 of the appendix A. It is important to note that although the GHZ state generation could be achieved by means of only three Bell pairs though the nodes, using four pairs yielded the best results as explained in the appendix.

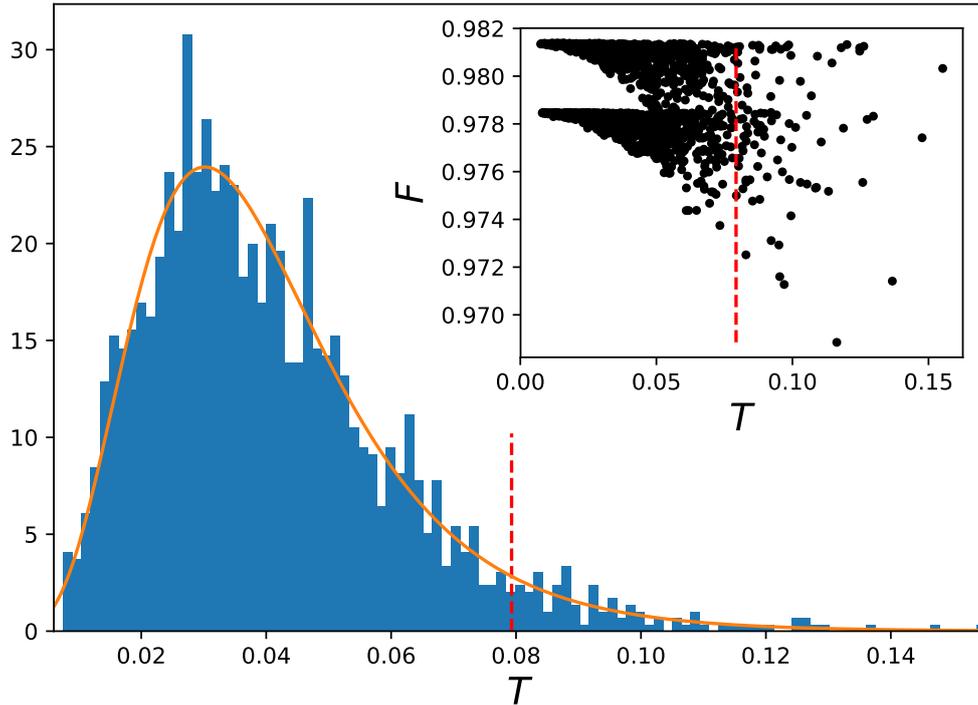


Fig. 4.3.: Histogram of the generation time for 2000 GHZ states using the improved parameters of table 4.1. The distribution is fitted to a skewed normal distribution, and a red line mark the 95% percentile for the faster times. (inset) Fidelity vs generation time for each GHZ state.

4.2.1 Dealing with indeterministic state generation

Since the GHZ generation process is not deterministic we need to deal with the probability that not all GHZ will be readily available at a given time. Thus, we need to introduce a cutoff that allows us to set a maximum time limit after which the unfinished states are ignored and the corresponding stabilizer measurement is not carried away. Fig 4.3 shows the fidelity vs time distribution for the generated entangled states, we see the distribution corresponds to skewed normal distribution. To set this maximum time we consider one in which the probability of obtaining a certain percentage of GHZ states is high enough. To take a pessimistic value we choose a time in which 95% of states have been completed.

On the surface code this process of ignoring certain stabilizer measurements can be seen as a classical erasure occurring on our system, in which the affected stabilizer will not perform any measurement and instead will report again the measurement result of the previous round. This is similar to adding additional measurement errors to the syndrome, in the sense that a classical erasure occurring on a stabilizer can give a false negative, namely fail to detect an error. Alternatively, it may also prolong any false positives originated from a previous true measurement

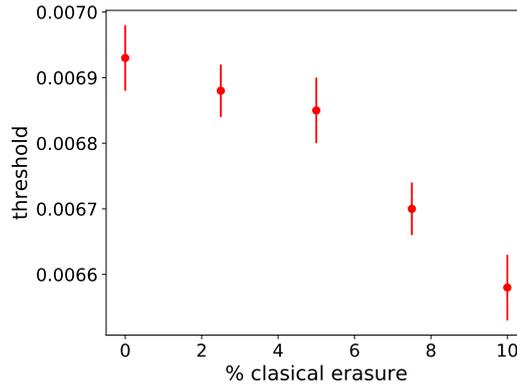


Fig. 4.4.: Thresholds for the monolithic architecture under the effect of classical erasure on the stabilizers, in which for each stabilizer measurement there is a probability of not taking place.

error, in other words a measurement error may propagate to the next round if the stabilizer fails to obtain new information. In order to deal with this new error we will consider the weights of the input graph of the decoder to be asymmetric, with double the value for physical error chains than for those of measurement error chains. This is a naive approach and should be considered non-optimal, for a better result the values of the weights should be optimized to increase the success chance of the decoder. With this in mind and the fact that our decoder is not suited to truly deal with classical erasure we may consider the threshold values obtained as lower bounds for the ones that can be achieved under an optimized decoder. We further discuss how can the decoder be improved in the outline section of this chapter.

As a consequence of the asymmetric weights we will use the logical error rate per measurement round, $p_l = P_l/N_{rounds}$ as defined in [35], when calculating the thresholds. This is to reflect the fact that since errors are more likely to occur in the measurements, we will require additional measurement rounds to counteract this. As a consequence, this more general definition of the error rate is required. Nonetheless, we should note that as discussed in [36] using this definition of error rate will give a small difference in the values of the thresholds to the ones obtained by using the error rate as defined in [18].

To find how this classical erasure affects the thresholds in our implementations, in fig 4.4 we see a comparison of how the threshold depends on the erasure percentage. We see that for low probabilities the effect is considerably small, where as for values above 7.5% the effectiveness starts to decay. We can understand this as when a stabilizer measurement is not finalized also the noise induced by the imperfect operations does not apply to the data qubits, and thus small classical erasure effects could potentially pass unnoticed. We must also note that we expect the effect of the classical erasure to be more detrimental as we increase the effect of time dependent memory induced errors.

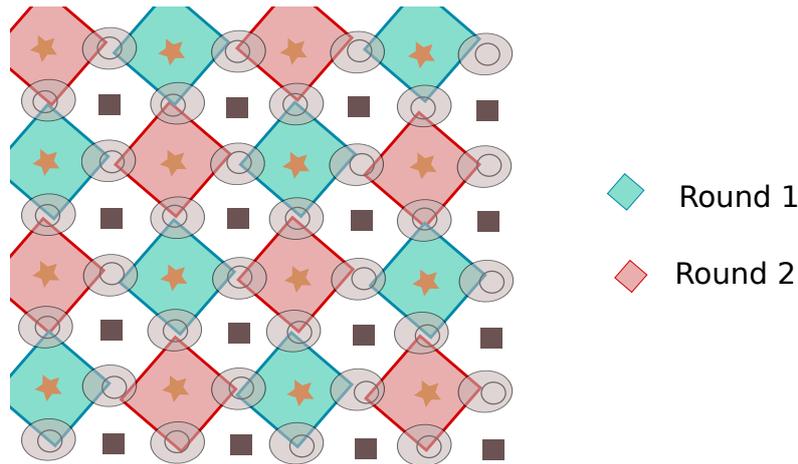


Fig. 4.5.: Example of measuring all the star stabilizers using the distributed architecture. Due to the limitations imposed by the entanglement generation two separate rounds are required to complete all measurements. The colors indicate how the stabilizers are taken for each round. This interspersed selection can analogously be applied to plaquette stabilizers.

4.2.2 Stabilizer measurement scheduling

We have seen previously that due to the nature of the methodology, we are limited to creating entanglement one pair of nodes at a time. This implies that when measuring one type of stabilizers across the entire surface code we must account for this limitation by scheduling the measurements in two parts. Consequently in total four rounds will be required to complete a round of measurements of all the stabilizers on the surface code. Figure 4.5 exemplifies how a complete round of star stabilizers measurements is carried.

The effect of requiring this two rounds for every stabilizer type will be reflected in an increase of time for every complete measurement, and consequently a increase also in the noise suffered by the data qubits. Ideally we would like to devise a method in which this two rounds are not required. As seen in chapter 5 this can be achieved by increasing the number of entanglement generation NV centers per node.

4.2.3 Additional protocols: Entanglement purification

Adding to the protocols for generating GHZ states, we will consider additional operations that perform entanglement distillation. Distillation aims to improve the fidelity of the multipartite state. As explained by [37] we can see entanglement distillation as a form of error correction over the entangled pairs. The general idea being that we consume one or more ancillary entangled pairs to increase the purity of a target state.

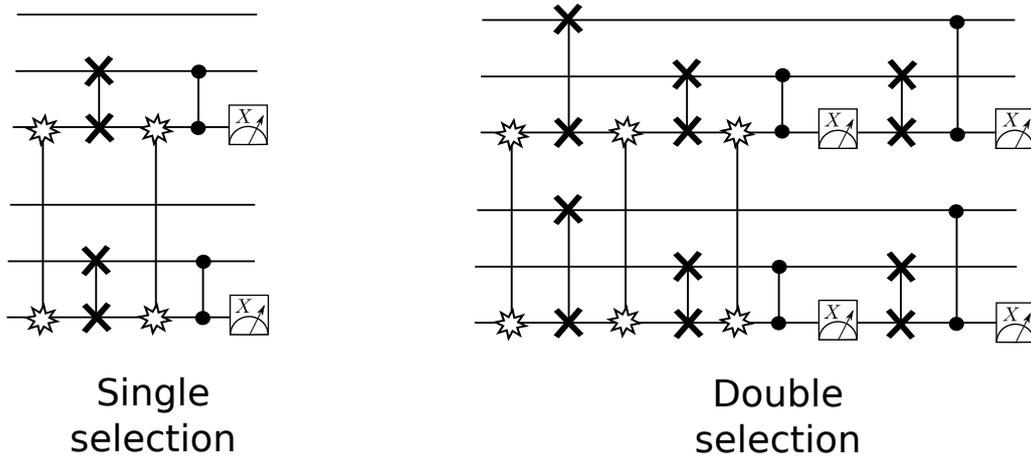


Fig. 4.6.: Purification protocols with all the required operations operations to work with the limitations of a NV center, for simplified diagrams see [4]. The result after each protocol is a Bell pair stored in the nuclear qubits. We select two different protocols of different complexities, a simpler protocol (right) requires two entangled pairs while a more complex one (left) requiring three pairs.

Following the work in [4] we consider two purification protocols of increasing complexity in the required operations for each one. In figure 4.6 we show the protocols considered with the exact operations involved in each one in the NV center context. It's worth mentioning that due to the limited number of qubits in each node, entanglement distillation is a process made over time. Namely, it involves the generation and storing of entangled states in order to perform the required operations. For our considered models this means that we need to be cautious about this extra required time as we have a trade-off between the purity gained by distillation and the extra noise added by the time dependent memory noise. In the next section we will see numerical comparison which shows how the effectiveness of this methods will be greatly hindered by this effects.

4.3 Simulating a quantum network

Now that we have established the theory, we need to numerically calculate the generation of the GHZ states through the nodes including the error models stated in chapter 3. To accomplish this, we make use of Monte Carlo simulations to create GHZ states. In order to reduce the computational time required we use an approach inspired by the event driven paradigm, in this case we simulate the entanglement generation as a event which requires multiple attempts before a success. Given the probability of success p , we simulate from success event to success event with the number of attempts required at each step obtained through drawing numbers

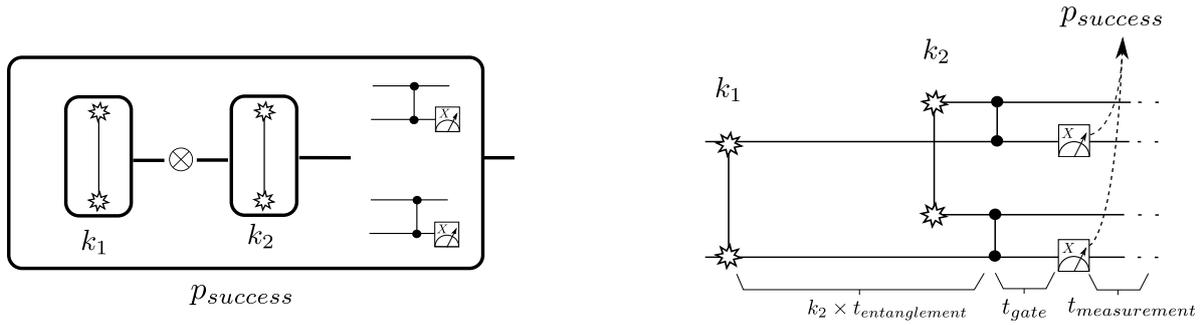


Fig. 4.7.: (left) Object oriented representation for entanglement purification. The circuits are assembled as objects (black boxes) which have an output and a success probability. Events can contain events within themselves required. From left to right lines connecting event symbolize the dependency on the success of the previous step. In this case the event of generating entanglement is simulated by drawing number k_1 and k_2 from the distribution. The entire process of purification has a probability of success which depends on k_1 and k_2 . (right) Circuit representation of the event oriented simulation. The number of attempts required in each step is proportional to the time taken by the circuit, visualized by the length of the channels between operations.

from the probability distribution of a set of independent boolean experiments, also known as a geometric distribution,

$$\Pr(X = k) = (1 - p)^k p; \quad k = 0, 1, 2, 3, \dots \quad (4.1)$$

This distribution describes the probability that the event succeeds after k failures given a event that outcomes only success/failure, with probabilities p and $1 - p$ respectively.

Given that we are dealing with probabilistic protocols, there is the probability that at some level the protocol fails and needs to be restarted. Then to create the simulations we treat each part of the circuit as an independent object, from which in order to advance to the next part the previous one had to be successful before. Following this method we can assemble the circuits as programming objects with each object containing the previous object inside. Figure 4.7 shows a diagram of how the circuits are assembled following the event oriented methodology. When running our simulation we run the circuit in a recursive manner, considering that in case of a failure at some level we need to restart that specific box starting from the lowest level, or from the inside out in terms of the objects. When completely assembled the result will be the simulation of one of the ensemble of circuits which generate GHZ states, we require to run this multiple times to calculate an average over the ensemble.

Parameter	Reference	Improved
p_s	0.006	0.003
p_g	0.006	0.003
p_m	0.006	0.003
a_0	83.3	8.0
a_1	$\frac{1}{3}$	$\frac{1}{30}$
η	$\frac{1}{500}$	$\frac{1}{100}$

Tab. 4.1.: Values for the parameters corresponding to the error models. Reference values according to the most recent literature for comparison, and improved values the actual values used though this work in the threshold calculations.

4.3.1 Protocols comparison

Now we present a numerical comparison between the different methods for creating a GHZ state. We propose 3 protocols to generate a fourpartite $|GHZ\rangle$ states among four nodes, each with different levels of complexity which show in the increase in fidelity and average creation time. The three protocols are as listed below:

1. EPL
2. EPL + Single selection
3. EPL + Double selection

Each protocol uses the single photon protocol as the basis for the raw entangled states. In each case 4 Bell pairs are used in order to generate the GHZ state. For the single and double selection protocols the EPL protocol is used as a basis for generating each ancilla pair used in the distillation (see fig. 4.6).

While we initially tested the parameters for the noise sources presented in section 3.4, we found that it was infeasible to generate useful GHZ states under such parameters. Being the maximum achievable fidelity of ≈ 0.8 , well above threshold in terms of the error induced. As such we propose the following set of improved parameters in table 4.1. Considering that to calculate thresholds we will fix all but one (or several depending in the case), and vary that parameter(s) until a threshold is found. The comparison of the average fidelities and generation times can be seen in the figure 4.8. We consider the parameters a_0, η , and p , with $p = p_s, p_g, p_m$, as a variable while keeping the rest fixed in each case. Additionally, we also considered using the BK protocol as the basis for the Bell pair generation. Although we don't show the results in fig. 4.8, we obtained the worst results by a great margin with the BK protocol in both resulting fidelity and

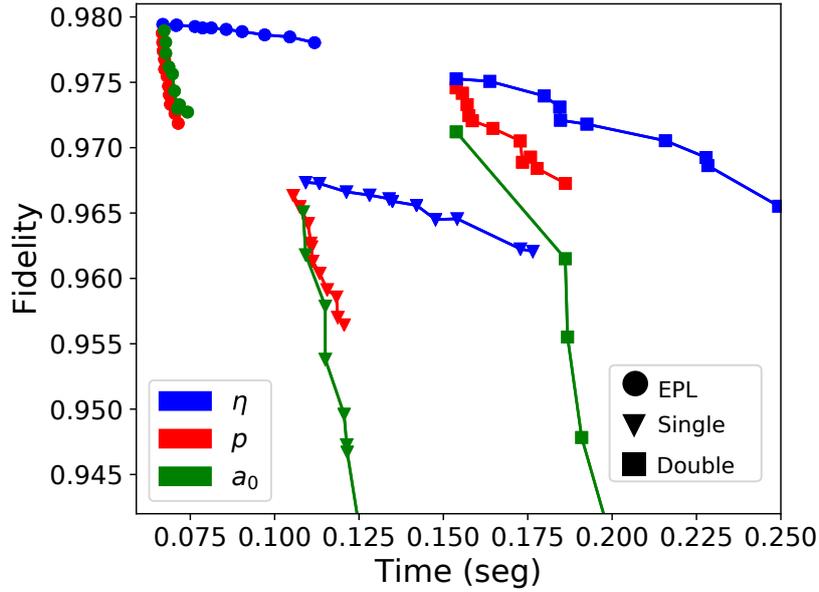


Fig. 4.8.: GHZ state fidelity vs time for each one of the protocols with the improved parameters stated above. As denoted by the colors, in three cases we keep constant all parameters and vary one: a_0 varying as $a_0 \in [5, 100]$, $\eta \in [0.01, 0.005]$ and $p \in [.003, .0041]$. For three different protocols shown using their respective symbols. In each case 2000 simulations were used to obtain an average.

generation times. The results show that the EPL protocol is the most effective at state generation, with both the highest fidelity and lower generation times. We can understand that the simplest protocol gives the best results is due to the presence of memory errors. Here time becomes a valuable resource and the time consuming distillation protocols induce more noise by storing the states than they counter by purifying. In consequence we will use the EPL protocol as the basis for generating GHZ when calculating thresholds for the rest of this work.

4.4 Effectiveness of the distributed implementation

Now that we have established which protocol is the most effective at generating entanglement. We will follow the methods exposed in section 2.3 in order to numerically calculate the thresholds over the distributed implementation. As discussed before we are interested in finding thresholds for the parameters a_0, η, p . Since we are interested in three parameters we can think of each obtained threshold, for a given set of values of said parameters, as a point in a surface over the parameter space. From an experimentalist point of view knowing the shape of this threshold surface would be of great convenience in order to assess the value his devices. Nonetheless, numerically this task is practicably intractable as it would require the manual calculation of

many points to be able to approximate the surface. Intuitively we can assume this surface to be smooth, non divergent and if parametrized by the error parameters to be strictly decreasing. Namely for any point in the surface if we move along the direction of increasing a parameter then the others must necessarily decrease and vice-versa.

As it is we will consider two cases: in the first scenario, which we will refer to as the *individual* parameter case, we will make use of the values presented in the table 4.1 for all the values expect the one in which the threshold is calculated. This scenario can be considered as a limit case when we analyze how noisy can be a specific part of our distributed quantum computer given the rest of the parts are of a high enough quality. In the second scenario, to be called the *combined* case, we will parametrize the three desired parameters by linear functions, such as:

$$\begin{aligned} a_0 &= f(\tau), \\ \eta &= g(\tau), \\ p &= h(\tau). \end{aligned}$$

With the functions given in detail in the appendix A, section 4. Here we find a threshold over the parameter τ and use plug this value into the functions to find the corresponding parameters. This will give us a middle ground threshold for all the values, which corresponds to a more realistic scenario of having all the devices contributing roughly equally to the noise induced into the system. In figure 4.9 we can found the graphs for each one of the thresholds obtained. The resulting fit gives the values for the individual case:

$$\begin{aligned} a_0^{ind} &= 43.9 \pm 2.0, \\ \eta^{ind} &= 0.0057 \pm 0.0002, \\ p^{ind} &= 0.00322 \pm 0.00002. \end{aligned}$$

And for the combined case by means of the parametrized functions we found:

$$\begin{aligned} a_0^{comb} &= 20.0 \pm 1.5, \\ \eta^{comb} &= 0.009 \pm 0.0002, \\ p^{comb} &= 0.0031 \pm 0.00002. \end{aligned}$$

Comparing the resulting threshold values with the reference values in table 4.1, we see that in general our results are way below this values. This thresholds give insights on the status of the current technology and which are the future possibilities for a distributed implementation of the surface code. As of now we can see that there are still great improvements to be made before

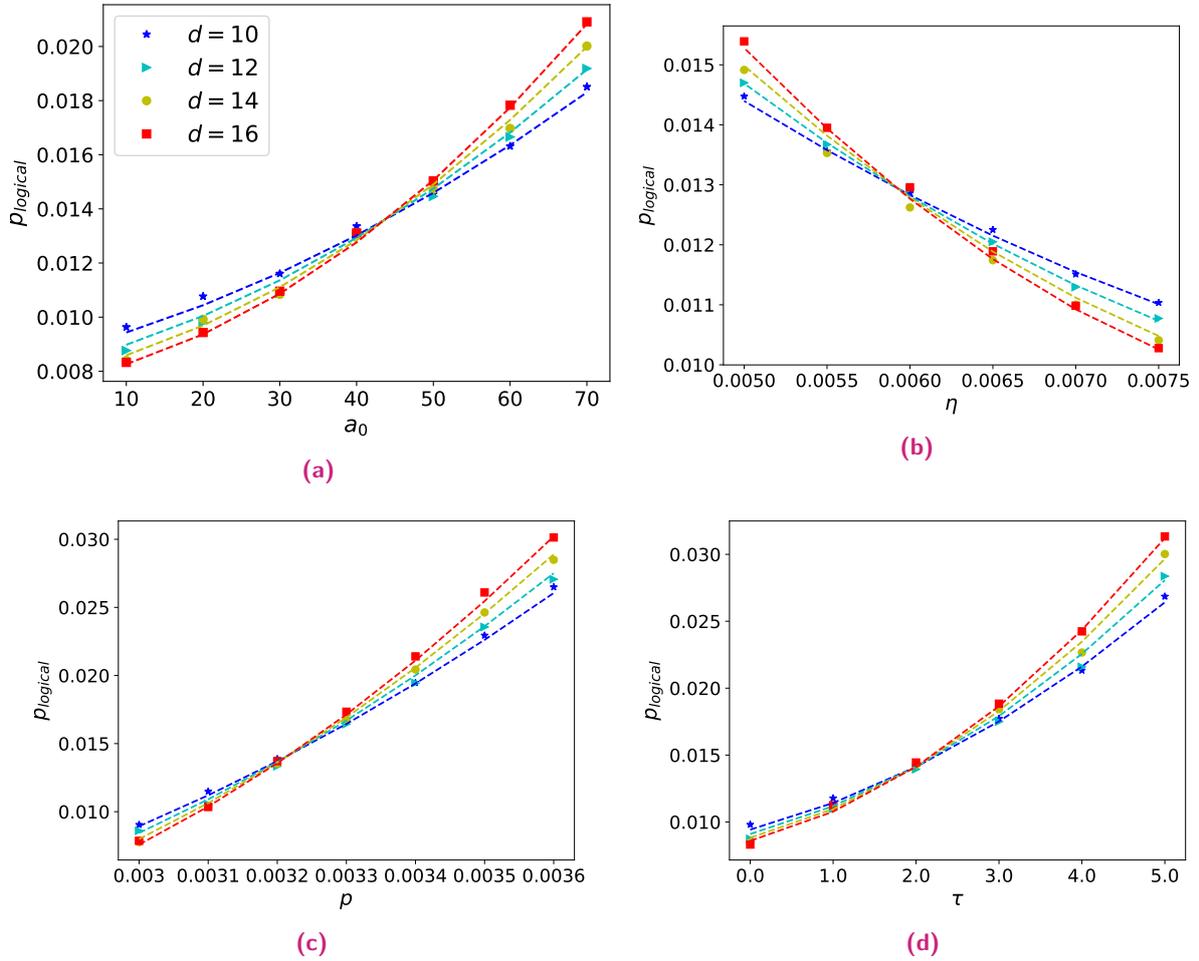


Fig. 4.9.: Thresholds for the distributed implementation under the improved parameters. In each case a number of 5×10^4 iterations are made to obtain p_l . Individual case: (a) Threshold over the noise induced by entanglement generation a_0 . (b) The threshold for the entanglement generation rate η . (c) Threshold for the noisy operations $p = p_s, p_g, p_m$. Combined case: (d) Thresholds varying the all parameters a_0, η and p by their corresponding parametrized linear functions.

we can consider creating a realistic implementation of this distributed quantum computer. In the next chapter we will show that the thresholds can be substantially raised by changing the scheme of how the distributed surface code is implemented.

4.5 Outline

We have seen that as a cause of using a non-deterministic method in evaluating the stabilizer measurements in the distributed implementation, we are inevitably introducing a classical erasure error into the surface code. This additional error should be accounted by altering the decoder to accommodate for this lack of information. Moreover, as shown in figure 4.8 we see how the great majority of GHZ states are completed within a relatively short time and thus an ideal decoder should take advantage of that. An asynchronous decoder which is capable to accept syndrome measurements as they become available while also dealing with classical erasure should shown a considerable increase in performance. This would effectively eliminate the decoherence incurred by the environment on data qubits while they are idle waiting for the entire sheet of stabilizer measurements. An example of a asynchronous decoder can be found in [38], where the authors make use of cellular-automata to perform the decoding.

Expanding network resources

With all considerations and methodology laid out in the previous sections, we showed some first thresholds with which we can assess the efficacy of the distributed implementation of the surface code using NV centers. Now we turn into exploring different manners in which the distributed implementation can be achieved that are more effective as a quantum memory.

In the first part of this chapter we increase the number of entanglement generation NV centers, to more efficiently generate multipartite states between the nodes. In the second part we further increase the capabilities of the network by increasing the number of data qubits hosted by each node. This allows for using lower weight GHZ states in the stabilizer measurements, weight denoting the number of qubits involved in the GHZ state, resulting in less noisy stabilizer measurements. In both cases we present the resulting thresholds and compare the results with each other and the previously obtained ones in chapter 4.

5.1 Parallel entanglement generation

We consider adding an extra NV center for entanglement generation to every node in the network. The goal of this extra NV center is to parallelize the entanglement generation process to reduce effects of the extra decoherence caused when entanglement is being generated. The idea is that parallel entanglement generation frees the process of the need to store an entangled pair while generating another one. Then we should see the difference over the decoherence corresponding to the parameter a_0 in our error model. Figure 5.1 shows a diagram of how the generation of a GHZ state under this new scheme, the circuit is analogous to the one of fig. 4.2. In this case we are considering that local two-qubit operations can be made between the two electron-spin qubits of the NV centers.

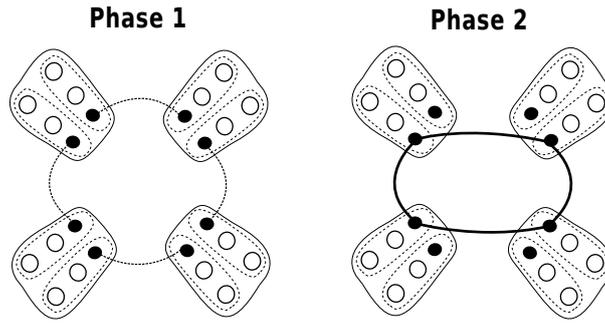


Fig. 5.1.: GHZ state generation with two entanglement generation dedicated NV centers per node, the operations involved are analogous to the ones in 4.2 with the first two phases reduced to one. In phase one all the required entanglement pairs are generated between the nodes. On the second phase local two-qubit operations are carried within the nodes and one qubit is measured to complete the GHZ state. As in the previous case, a correction operator may or may not be necessary depending on the outcome of the measurements.

The resulting thresholds obtained under this architecture, as shown in fig. 5.2 are; first for the individual case:

$$\begin{aligned}
 a_0^{ind} &= 3870.0 \pm 128.2, \\
 \eta^{ind} &= 0.0028 \pm 0.0002, \\
 p^{ind} &= 0.00332 \pm 0.00002.
 \end{aligned}$$

And for the combined parameter scenario:

$$\begin{aligned}
 a_0^{comb} &= 440.0 \pm 60, \\
 \eta^{comb} &= 0.0072 \pm 0.0002, \\
 p^{comb} &= 0.00319 \pm 0.00002.
 \end{aligned}$$

We see that the most drastic change occurs over the threshold in a_0 in the individual case, this new value is several orders of magnitude greater than in the previous case. This effectively shows how we can almost nullify the decoherence effects of generating entanglement by using this method. For the other parameters we see a modest improvement in the values of the threshold although not as drastic.

Although there are other possible ways to use this extra NV center, for example to perform entanglement distillation, we have not tested the results of these different approaches. However we expect that different approaches would still be significantly sensitive to the effects of the decoherence induced when generating entanglement.

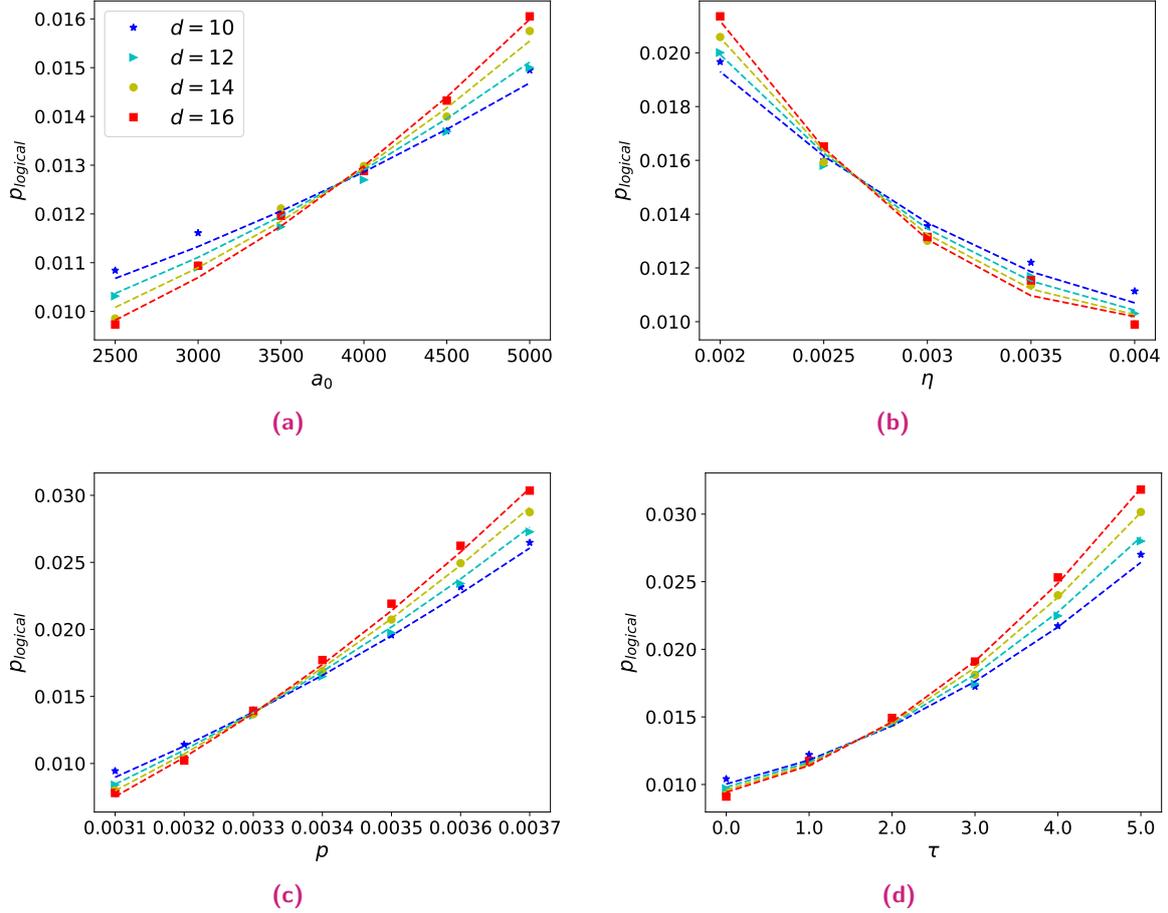


Fig. 5.2.: Thresholds for the distributed implementation under parallel entanglement generation. (a) The threshold for a_0 increased by two orders of magnitude when using this new methodology there are no stored entangled pairs which can decohere when generating additional pairs. The thresholds for η (b) and p however show only a slight improvement over the previous values. In (d) we shown the threshold for the parameters in the combined scenrio.

5.2 Clustering data qubits

Now we are interested in a different type of architectures for the network. Further increasing the resources of the network, we can devise new surface code schemes with more than one data qubit per node. The idea is that if we can place more than one data qubit per node we can reduce the weight of the GHZ state required for the stabilizer measurements, as mentioned in the previous chapter. For this new architectures we require to consider nodes which contain additional NV centers for entanglement generation, as well as additional connections between the nodes. In this new network each nodes connects to their diagonal neighbors in addition to the nearest neighbor connections.

With this more robust network we will consider two schemes with two and three data qubits per node. We aim at reducing the weight in the shared GHZ state by having two or more qubits sharing the same ancilla locally. This reduction on the weight of the GHZ states greatly reduces the noise introduced by all the network operations. Nonetheless, we must note that generating entanglement between nodes is a pairwise operation (entanglement generation can only be made between two nodes at the same time), meaning that the larger the nodes we will need additional entanglement generation capabilities to counter this limitation.

Figure 5.3 shows the network construction with two data qubits per node. In which 6 communication NV centers plus the NV center with 2 nuclear qubits as data qubits are required by each node. While fig. 5.4 shows the network for three data qubits per node, which requires 7 communication NV centers and one additional with 3 nuclear qubits. We can see that in the hybrid scheme we require more NV centers for entanglement generation due to the loss of parallelization incurred when further clustering the data qubits. Is also worth mentioning that when including more than one local operations over a single node we are effectively introducing some correlated noise coming from the noisy network. Although the scope of these errors is not studied in this work, previous work in [35] consider this problem. They found that by giving the MWPM decoding extra information by altering the weights in the graph used the effect of this correlated noise can be reduced and higher thresholds can be achieved.

The resulting threshold values for the architectures are shown in figures 5.5 for the paired architecture, in the individual parameter scenario:

$$\begin{aligned}\eta^{ind} &= 0.00077 \pm 0.00007, \\ p^{ind} &= 0.00361 \pm 0.00002.\end{aligned}$$

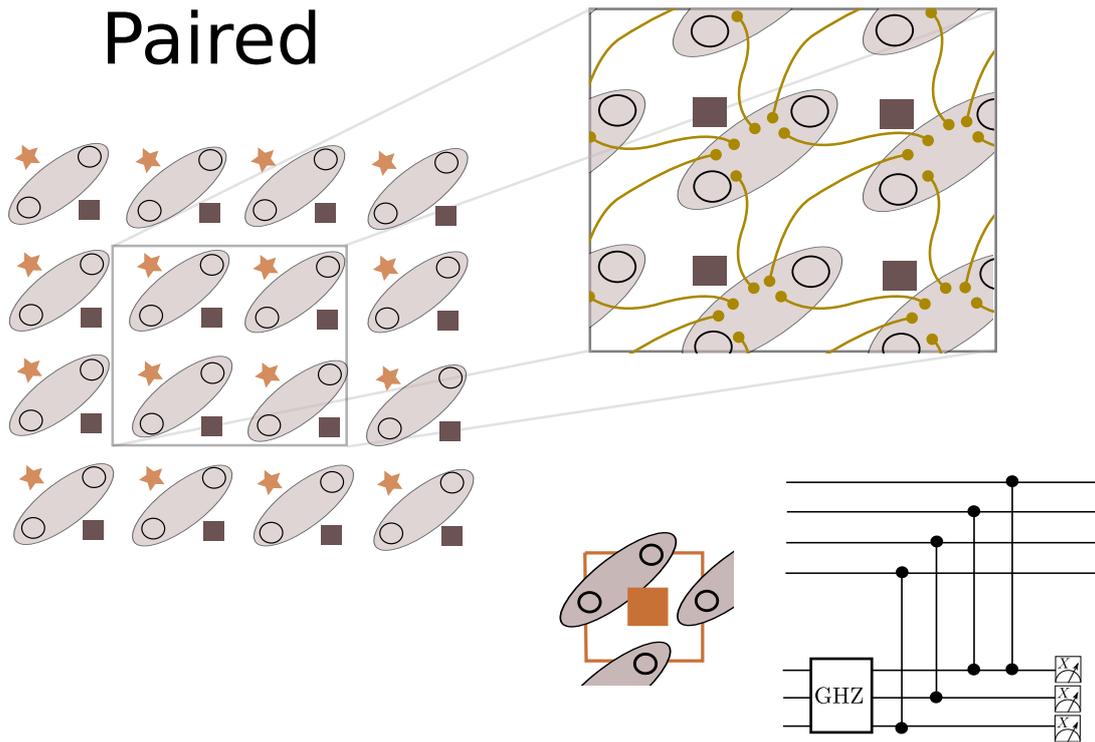


Fig. 5.3.: Paired scheme in which two data qubits are considered in every node. This allows to reduce the weight of the required GHZ state in the stabilizer measurements to a tripartite state. To be able to generate all the required states to evaluate all the stabilizers in one step 7 NV centers are required in each node.

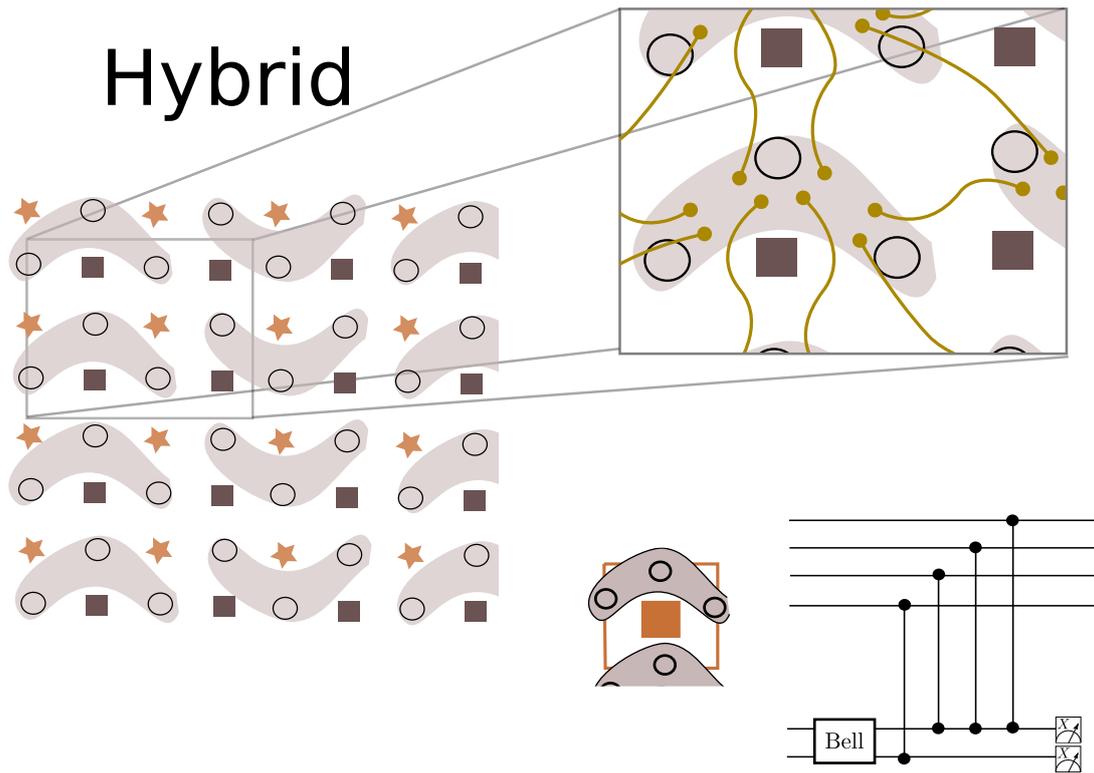


Fig. 5.4.: Hybrid scheme with three data qubits per node. Using this scheme 1/3 of the stabilizer measurements can be achieved by a two-partite state, thus further minimizing the error introduced by the noisy stabilizer measurements. Each node requires a total of 9 NV centers to be able to evaluate all stabilizers in one step.

For the combined parameter scenarios, using the values of the functions specified as in the appendix A, are:

$$\begin{aligned} a_0^{comb} &= 100.0, \\ \eta^{comb} &= 0.0048 \pm 0.0002, \\ p^{comb} &= 0.00352 \pm 0.00002. \end{aligned}$$

Now for the hybrid scheme the resulting error rates around the threshold are shown in fig. 5.6. The results obtained are for the individual scenario:

$$\begin{aligned} \eta^{ind} &= 0.00047 \pm 0.00002, \\ p^{ind} &= 0.00427 \pm 0.00002. \end{aligned}$$

Lastly, for the combined parameter case the resulting values are:

$$\begin{aligned} a_0^{comb} &= 100.0, \\ \eta^{comb} &= 0.0059 \pm 0.0002, \\ p^{comb} &= 0.00421 \pm 0.00002. \end{aligned}$$

We see in both cases a great increase in the resulting values when compared with the previous ones. In the individual case of the parameter p we see that the hybrid scheme specially presents a great improvement over the other cases. This is understandable as by reducing the weight of the GHZ state we are directly diminishing the noise caused by the imperfect operations. Nonetheless, if we compare the values obtained to the ones in the references, we see the ones we have obtained are considerably lower. This can be understood when considering the hardware limitations of the NV centers which necessitate noisy SWAP operations frequently which magnifies the effects of noisy operations in our implementation. For the transmissivity parameter η in the individual case, we see each new scheme provides good improvements, with the best result an order of magnitude better than the first implementation and well within range of the current experimental value. Nonetheless, here we must also consider that although the thresholds are quite high the practical time it takes for evaluating a complete round of stabilizers is also quite high, around 2 seconds. Meaning that potentially the computing times could easily become intractable in a real application in which several hundreds of measurement rounds are required.

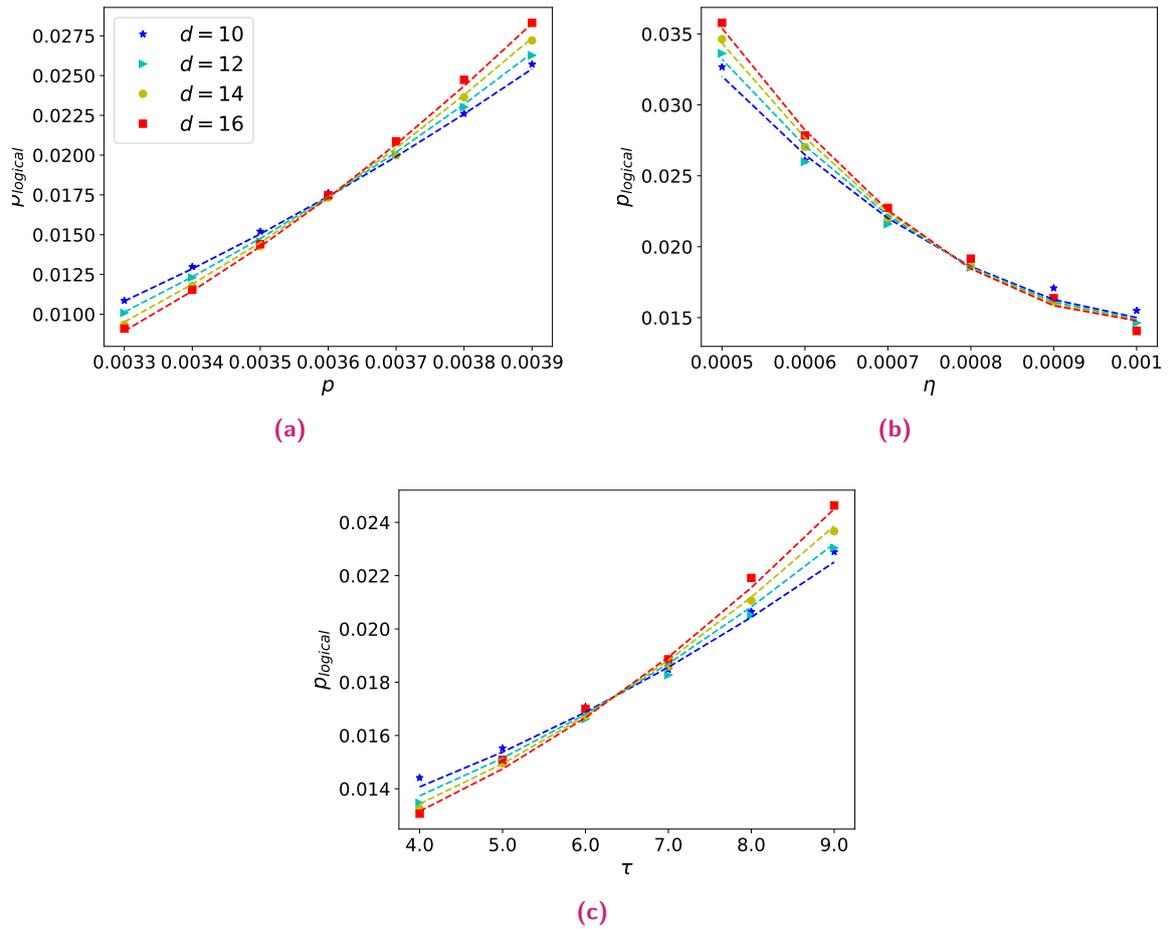


Fig. 5.5.: Thresholds results for the paired scheme. In the individual scenario the threshold for the parameter p (a) and for the parameter η (b). In the combined parameter scenario the obtained threshold over τ (c), for the corresponding set of functions.

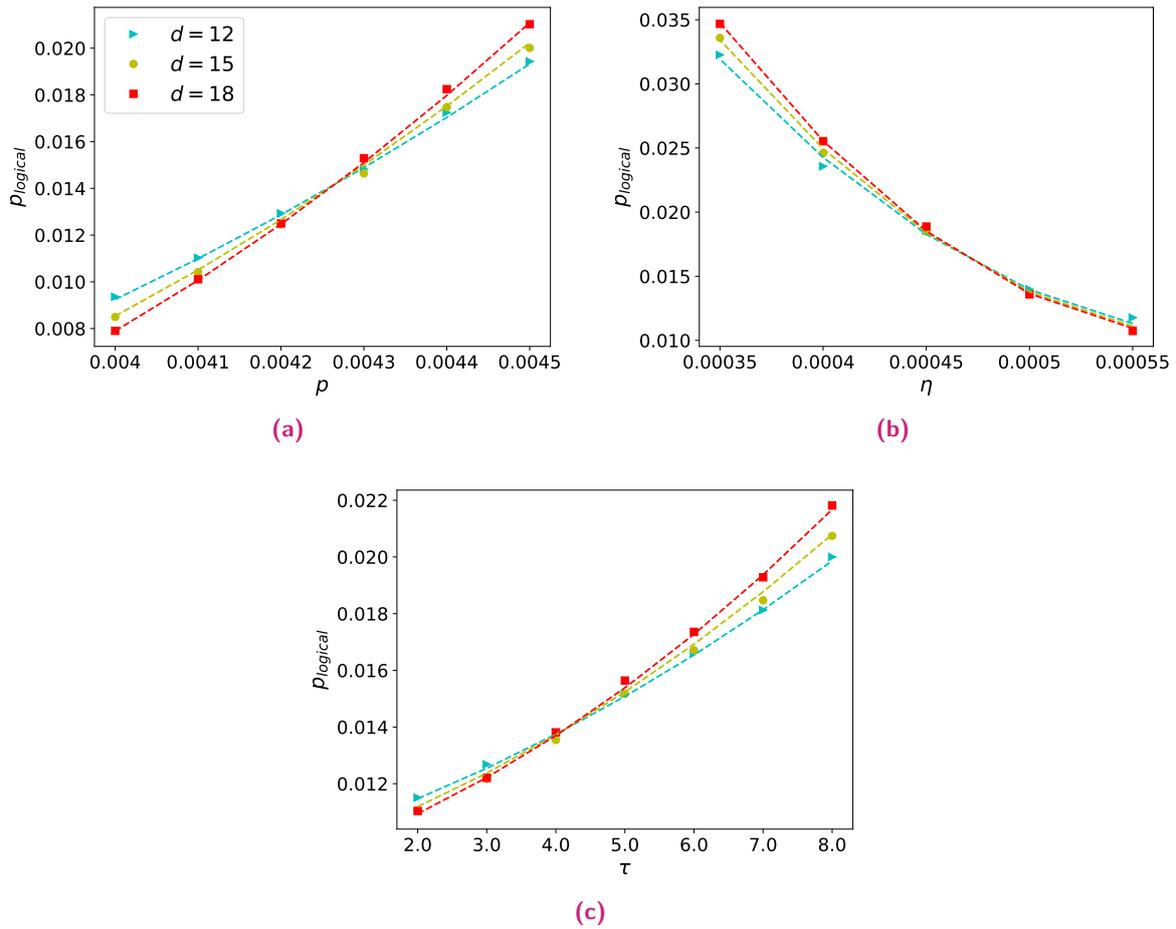


Fig. 5.6.: Thresholds in the hybrid architecture. For the individual case results on the parameters p (a) and η (b). For the combined scenario the resulting threshold over τ (c), by means of its specific set of parameter functions.

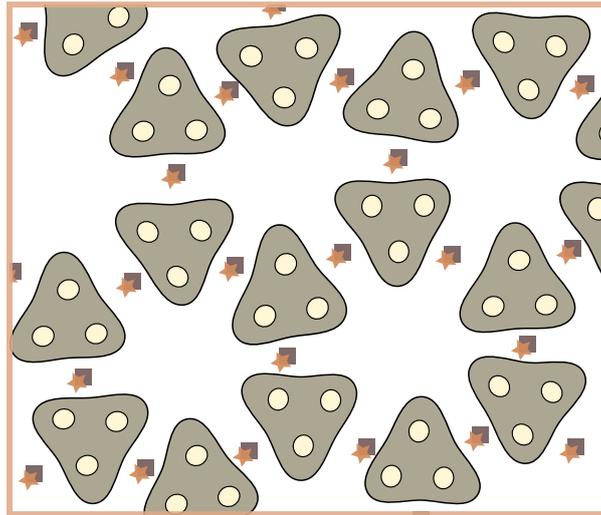


Fig. 5.7.: Schematic of a topological error correcting code designed specifically for quantum networks. By adding additional connections in the nodes and clustering the data qubits, all stabilizer measurements can be achieved by shared Bell pairs between the nodes.

5.3 Outline

We consider that further work in distributed quantum computing could be done in designing special error correcting codes for distributed architectures. Specifically with the previous results in mind, we consider that a good starting point would be by adding additional connections to the surface code while clustering the data qubits in order to reduce the weight of the entangled states required. Figure 5.7 shows a schematic of a surface with 3 data qubits per node and additional connections between each node in which all stabilizer measurements can be achieved by a shared Bell pair between nodes. However, we have not fully tested this device and further research into this or other possible constructions must still be carried in order to truly determine the effectiveness of this type of designs.

Conclusions

We studied a distributed implementation of the surface code motivated by previous works and the recent development of NV centers as quantum computing devices. Our analysis considered realistic error models which cover the main sources of noise in a quantum network composed of NV centers. For the first part we compared different protocols for generating the shared GHZ states between nodes. The results show that the simplest protocols yielded the best results, with the EPL protocol resulting in the best state fidelities and shorter generation times. This is a consequence of the consideration of the times required to generate entanglement between nodes which introduces a time dependent noise induced by the environment on the qubits inside a NV center. For the limit in which entanglement can be generated very swiftly we expect to see a different behavior in which indeed more complex protocols using entanglement distillation would provide better results, as seen in [5].

To assess the effectiveness of the distributed surface code we calculated the thresholds for the most relevant noise parameters in two scenarios. A first one in which we increase the error rate of a single noise source while considering that the rest of the noise sources are considerably small. In a second scenario we varied the most relevant parameters of noise effects simultaneously in order to find a middle ground set of thresholds for the selected sources. We can consider the insights given by the first scenario as a limit case in which we ask ourselves how noisy can a specific part of our distributed surface code can be, given the remaining parts are good enough. Conversely we can picture the second scenario as a more realistic in which all parts contribute roughly equally to the over all failure rate of the code.

For our first implementation of the distributed surface code with NV center, the thresholds calculated still differ considerably with the experimental values obtained in the laboratory, shown in table 4.1. This shows that improvements must still be made in order to make a practical construction of a distributed surface code with NV centers. Nonetheless, we obtained understanding around the effects of different noise sources, and more importantly we saw that there is still room for improvements which could allow for a more successful implementation. Specifically, we mentioned how the implementation could greatly benefit from having an asynchronous decoder. Using this decoder could essentially eliminate the idle times incurred by the nodes during the entanglement generation, consequently greatly reducing the time dependent decoherence effects.

In order to circumvent the noise incurred into the surface code by the hardware we considered introducing additional resources into the network in order to devise new architectures which give better results. A first architecture contemplates an additional NV center in each node in order to generate entanglement in parallel and thus decrease the effects of the decoherence induced by the entanglement generation method. The results show some moderate improvement in the parameters with the exception of the parameter a_0 associated to the aforementioned noise effect, which increases by two orders of magnitude in comparison with the initial implementation. This shows that this effect can be effectively disregarded under this architecture.

Furthermore, we considered an additional approach which consisted on clustering the data qubits of the surface code into the nodes. Using these architectures relaxes the restrictions on the weight of the GHZ states involved on the stabilizer measurements, which can now be lower depending on the specific construction. We proposed two architectures, one which reduces the weight of all the GHZ states to three, and a second which further permits for one third of the stabilizer measurements to be carried with only Bell pairs. Both architectures display considerable improvements on the thresholds values, with the threshold of the transitivity parameter η increasing by an order of magnitude, bringing it into the range of current experimental values obtained in [2]. With respect with the operation fidelity parameter p , we saw that although the threshold showed an increase of $\approx 30\%$, the values still fall short of the ones obtained in previous works of distributed implementations [5]. We believe the reason for this is due to the requirement of additional operations, such as the extra SWAP operations used to overcome the hardware restriction intrinsic to NV centers. However, we believe these results point towards directions where further development of distributed quantum computing could be made, and motivate investigation on specially designed error correcting codes for quantum computing over a network.

Appendix A: Definitions and derivations

A.1 Fidelity

We define the fidelity between two quantum states ρ and σ as:

$$F(\rho, \sigma) = \text{Tr} \left[\sqrt{\sqrt{\sigma} \rho \sqrt{\sigma}} \right].$$

In the case when one of the states corresponds to a pure state $\sigma = |\psi\rangle\langle\psi|$ the definition of fidelity is reduced to

$$F(\rho, |\psi\rangle) = \sqrt{\langle\psi| \rho |\psi\rangle}.$$

A.2 State twirling

A twirling operation on the state can be seen as a random application of SWAP gates over the qubits of the state until the state mixture is equally distributed over all the qubits. Formally we define the twirling operation \mathcal{T} as:

$$\mathcal{T}(\rho) = \frac{1}{d} \sum_{P_i \in P(n)} P_i \rho P_i^\dagger,$$

with $P(n)$ the set of d distinct permutations over n qubits.

$b_1b_2b_3b_4$	correction
0000	I
0011	I
1100	I
1111	I
0101	XX
1010	XX
1001	XX
0110	XX

Tab. A.1.: Correction operators required depending on the resulting bit of the measurements b_i during the GHZ creation protocol. When using four entangled pairs, four measurements are required.

A.3 GHZ state generation

Now we present the calculation involved in producing a GHZ state from 4 Bell states $|\phi^+\rangle$ shared between nodes A, B, C and D . Where for simplicity we perform the calculation using pure states, but the same methodology applies for the mixed state case.

$$\begin{aligned}
|\Psi\rangle &= |\phi^+\rangle_{AB} |\phi^+\rangle_{CD} |\phi^+\rangle_{BD} |\phi^+\rangle_{AC} \\
&= \left[\frac{1}{4} |00000000\rangle + |00000011\rangle + |00001100\rangle + |00001111\rangle + |00110000\rangle + |00110011\rangle + \right. \\
&\quad |00111100\rangle + |00111111\rangle + |11000000\rangle + |11000011\rangle + |11001100\rangle + |11001111\rangle + \\
&\quad \left. |11110000\rangle + |11110011\rangle + |1111111100\rangle + |11111111\rangle \right]
\end{aligned}$$

Applying the corresponding local $CNOT$ gates in each node.

$$\begin{aligned}
|\Psi\rangle &= \frac{1}{4} \left[|00000000\rangle + |00000011\rangle + |00001100\rangle + |00001111\rangle + |00110101\rangle + |00110110\rangle + \right. \\
&\quad |00111100\rangle + |00111101\rangle + |1001010\rangle + |11001001\rangle + |11000110\rangle + |11000101\rangle + \\
&\quad \left. |11111111\rangle + |11111100\rangle + |1111110011\rangle + |11110000\rangle \right]
\end{aligned}$$

Where we measure the last four qubits and apply a correction operation following the result of the obtained measurements to obtain the GHZ state. The table A.1 shows the possible measurement results and if any the required correction.

From the table we can see that we are also doing step of state distillation, in which we are only keeping the resulting state if the resulting bit string has a even Hamming weight. If the resulting string had an odd weight that means the state collapsed into the incorrect subspace and is no

$b_1 b_2$	correction
00	\mathbb{I}
11	\mathbb{I}
01	XX
10	XX

Tab. A.2.: Correction operators required depending on the resulting bit of the measurements b_i for the special case when three entangled pairs are used.

longer useful. Also we can see that this extra step of distillation would not be possible if we would only use three Bell pairs to form the GHZ in which case the dimension of the ancilla Hilbert space is not big enough for the state to go into either even or odd parity. We can see this in table A.2, where we show all the possible measurement results for the case when three pairs are used. In this case there is no extra room for the resulting state to collapse to a given parity. The resulting GHZ would still be a mixture of both parities and consequently it would introduce more noise to the data qubits during the stabilizer measurements.

A.4 Parametrization over the parameter space

In this section we will describe the functions used to vary the three parameters a_0, η and p simultaneously, in order to calculate a threshold over the three of them. The parameters are varied through the parametrized functions of τ , with:

$$\begin{aligned} a_0 &= f_i(\tau), \\ \eta &= g_i(\tau), \\ p &= h_i(\tau). \end{aligned}$$

For each different architecture we will consider different set of functions and the parametrization over $\tau = [0, 9]$ in increments of 1. For each implementation with the threshold values are reported as shown in figures 4.9, 5.2, 5.5 and 5.6 respectively with the following functions:

- **Initial distributed implementation**

$$\begin{aligned} f_1(\tau) &= 10. + 5.\tau, \\ g_1(\tau) &= 0.01 - 0.0005\tau, \\ h_1(\tau) &= 0.003 + 0.00005\tau. \end{aligned}$$

- **Parallel entanglement generation**

$$\begin{aligned}f_2(\tau) &= 100.0 + 200.\tau, \\g_2(\tau) &= 0.008 - 0.0005\tau, \\h_2(\tau) &= 0.0031 + 0.00005\tau\end{aligned}$$

- **Paired architecture**

$$\begin{aligned}f_3(\tau) &= 100.0, \\g_3(\tau) &= 0.008 - 0.0005\tau, \\h_3(\tau) &= 0.0032 + 0.00005\tau.\end{aligned}$$

- **Hybrid architecture**

$$\begin{aligned}f_4(\tau) &= 100.0, \\g_4(\tau) &= 0.008 - 0.0005\tau, \\h_4(\tau) &= 0.0040 + 0.00005\tau.\end{aligned}$$

Appendix: Noise modeling

In this appendix we will state all the mathematical formalisms and the specific procedure used into modeling the errors that come from performing a noisy stabilizer measurement.

B.1 The χ -matrix

Every quantum process \mathcal{E} can be expressed in terms of the Kraus decomposition,

$$\mathcal{E}(\rho) = \sum_i^n K_i \rho K_i^\dagger.$$

Where K_i are the Kraus operators. In general the Kraus representation is not unique, a different set of operators could be used to obtain the same process. Nonetheless, it can be shown that all of them are equivalent up to a unitary operation [39]. The minimum number of Kraus operators that can be used is called the rank of the process. In the special case with the rank is 1 the process is called pure, and we can see that if \mathcal{E} is both completely positive and trace preserving (CTTP) then the process is simply a unitary evolution.

In general we can use any arbitrary basis to describe our Kraus operators as

$$K_i = \sum_n a_{in} \Upsilon_n$$

with Υ_n matrices of dimension $d \times d \in \mathbb{C}$. Then the channel expressed in this basis is

$$\mathcal{E}(\rho) = \sum_i^{d^2} \chi_{m,n} \Upsilon_m \rho \Upsilon_n^\dagger.$$

Where the values of the basis are accounted as $\chi_{m,n} = \sum_i a_{mi} a_{ni}^*$. This $d^2 \times d^2$ matrix called χ -matrix is very important since it holds all the information about the quantum process. We can see that nonetheless this matrix depends on the basis used to describe our quantum process. However is not difficult to see that we can make a change of basis $\chi' = U_0 \chi U_0^\dagger$, with U_0 a unitary operator, in order to obtain all the information in a different basis.

B.1.1 The Choi-Jamiolkowski isomorphism

Mathematically the Choi-Jamiolkowski isomorphism consists in a very simple transformation of all the mathematical objects we use to express states in quantum mechanics [40]. Essentially, it says that we must make the following equivalence,

$$|i\rangle\langle j| \equiv |i\rangle|j\rangle.$$

That is matrices are changed into vectors and superoperators are changed into operators. Physically this equivalence carries a very strong idea, the idea that mixed states can be seen as partial traces of pure states on higher dimensions. Analogously it tells us every non-unitary quantum operation can be seen has a unitary one on a higher dimensional space. Performing the isomorphism on a mathematical object directly points to this space, up to some constant. The meaning of this equivalence appears to be that every object or operation can be seen as a (part of a) fully quantum object if we look at a big enough Hilbert space.

With this isomorphism in mind we can redefine our basis in which the Kraus operators are expressed. The matrix vector equivalence in the basis is expressed as:

$$\Upsilon_n \equiv |\Upsilon_n\rangle\rangle, \quad \Upsilon_n \equiv \langle\langle \Upsilon_n|.$$

Then we can write the channel transformed into an operator,

$$\hat{\mathcal{E}} = \sum_{m,n} \chi_{m,n} |\Upsilon_m\rangle\rangle \langle\langle \Upsilon_n|.$$

Thus, we see that by performing the isomorphism we can recover all the information of the process, the χ -matrix, automatically.

Now the clear question is how can we do this in practice. Certainly it is impossible to alter an experiment in order to change all matrices into vectors. Nonetheless, there is a simple way to use the isomorphism through what is commonly referred as ancilla assisted operations. First, we define the bipartite maximally entangled state:

$$|\Phi\rangle = \frac{1}{\sqrt{d}} \sum_i^d |i\rangle \otimes |i\rangle.$$

On which we apply the channel to one half of the state to obtain a density matrix, also called Choi matrix:

$$\begin{aligned}\rho_{\Phi} &= (\mathcal{E} \otimes \mathbb{I})(|\Phi\rangle\langle\Phi|) = \frac{1}{d} \sum_{i,j} \mathcal{E}(|i\rangle\langle j|) \otimes |i\rangle\langle j| \\ &= \frac{1}{d} \sum_{i,j} \left(\sum_{m,n} \chi_{m,n} \Upsilon_m |i\rangle\langle j| \Upsilon_n^\dagger \right) \otimes |i\rangle\langle j|\end{aligned}$$

Where if we examine the action on the channel we may recognize that all the required information is already there. However, is not completely clear how to obtain the χ -matrix from ρ_{Φ} . To this end, we can define the basis vectors as the matrices Υ_i applied to the bipartite state.

$$\begin{aligned}|\Upsilon_n^\Phi\rangle\rangle &= (\Upsilon_n \otimes \mathbb{I}) |\Phi\rangle \\ &= \frac{1}{\sqrt{d}} \sum_i \Upsilon_n |i\rangle \otimes |i\rangle.\end{aligned}$$

With this basis in mind its easy to see, by rearranging the sums, how the density matrix is expressed in terms of these vectors.

$$\begin{aligned}\rho_{\Phi} &= \frac{1}{d} \sum_{m,n} \chi_{m,n} \left(\sum_i \Upsilon_m |i\rangle \otimes |i\rangle \right) \left(\sum_j \langle j| \otimes \langle j| \Upsilon_n^\dagger \right) \\ &= \sum_{m,n} \chi_{m,n} |\Upsilon_m^\Phi\rangle\rangle \langle\langle \Upsilon_n^\Phi|\end{aligned}$$

Where we see that all the process information is inside this density matrix encoded in the specific basis we choose to create the vectors.

B.2 Noisy stabilizer modeling

With the mathematical prelude above we can formulate how to model in general every noisy operation through the χ -matrix. Now we want to examine the problem for the specific case when the channel is a noisy stabilizer. To achieve this we need to choose a specific basis that will allow us to model the channel through the probabilities of all the possible errors occurring. Here we specify step by step the exact process in which we construct such basis and extract the corresponding probabilities in order to formulate a complete model of the channel.

1. Identify the stabilizer as a channel. A stabilizer includes a measurement in some ancilla space that distinguishes our resulting states into even and odd parities according to a specific parity, (X for stars and Z for plaquettes).

$$\mathcal{S}(\rho_{in}) = p_{even}\rho_{out}^{even} \otimes |0\rangle\langle 0| + p_{odd}\rho_{out}^{odd} \otimes |1\rangle\langle 1|$$

The extra registry contains the information about the measurement result with the corresponding probability.

2. Do the isomorphism. Using a bipartite state we apply the channel to a half to obtain the χ -matrix.

$$\rho_{\Phi} = (\mathcal{S} \otimes \mathbb{I})(|\Phi\rangle\langle\Phi|)$$

3. Define a basis to analyze ρ_{Φ} . We wish to express the χ -matrix into a useful basis that will allow us to extract the possibilities of the errors that can occur into a noisy stabilizer. First, we will use the Pauli basis $\mathcal{P} = \{P^{\otimes d}\} \otimes \mathbb{I}$, that is the combination of all the possible Pauli matrices $P = \mathbb{I}, X, Y, Z$ acting on a single side of the bipartite state. This will allow us obtain the probabilities of different Pauli errors occurring in the data qubits measured by the stabilizer.

Additionally we need to model the possibility of a measurement error happening, in other words, when the returned registry of the measurement is incorrect. To account for this we must distinguish between both even and odd states. For this we use the ideal projectors into both subspaces.

$$|\Phi^{even}\rangle = (\Pi_{even} \otimes \mathbb{I}) |\Phi\rangle, \quad |\Phi^{odd}\rangle = (\Pi_{odd} \otimes \mathbb{I}) |\Phi\rangle$$

Were the projectors into the even or odd parities into a basis are defined as a combination of operators corresponding the basis in which the parity projection is made applied to the corresponding qubits. For example into the Z parity:

$$\Pi_{even} = \frac{1}{\sqrt{2}}(\mathbb{I} + \bigotimes_i^d Z), \quad \Pi_{odd} = \frac{1}{\sqrt{2}}(\mathbb{I} - \bigotimes_i^d Z).$$

Then one vector in the new basis for a given parity and Pauli operator $E \in \mathcal{P}$ is given by:

$$\begin{aligned} |\Phi_E^{OK}\rangle &= E |\Phi^{even}\rangle \otimes |0\rangle + E |\Phi^{odd}\rangle \otimes |1\rangle, \\ |\Phi_E^{NOK}\rangle &= E |\Phi^{odd}\rangle \otimes |0\rangle + E |\Phi^{even}\rangle \otimes |1\rangle \end{aligned}$$

Where it is important to realize that since we are doing a parity projection, there is going to be a certain symmetry with respect to this parity. For example if the parity projected is Z then we can see that every state is also an eigenstate of $\bigotimes_i^d Z$, $(\bigotimes_i^d Z)\Pi_{(even,odd)} = \pm\Pi_{(even,odd)}$, with eigenvalues 1 or -1 depending in the parity of the state. This means that there is an equivalence into the set of the Pauli basis, for this case $E \equiv \bigotimes_i^d Z E$. This means we will only use a subset of the Pauli operators when analyzing the model. Finally, the complete basis is given by all the states $\{|\Phi_E^S\rangle\}$. With $S \in \{OK, NOK\}$ Where the probabilities for each error are given as:

$$p_E^S = \langle \Phi_E^S | \rho_\Phi | \Phi_E^S \rangle$$

As a side note we must see that there exists an equivalence between the projectors and the Pauli errors E . That is when one permutes the operations E and $\Pi_{(even, odd)}$ then one obtains a different vector of the basis defined. For example let $E = X \otimes \mathbb{I}$ then:

$$\begin{aligned} (X \otimes \mathbb{I})\Pi_{\text{even}} &= \frac{1}{2}(X \otimes \mathbb{I})(\mathbb{I} + \bigotimes_i^d Z) \\ &= \frac{1}{2}\mathbb{I}(X \otimes \mathbb{I}) - \bigotimes_i^d Z(X \otimes \mathbb{I}) \\ &= \frac{1}{2}(\mathbb{I} - \bigotimes_i^d Z)(X \otimes \mathbb{I}) \\ &= \Pi_{\text{odd}}(X \otimes \mathbb{I}). \end{aligned}$$

Thus the general analysis if we were to change the order of this operations when doing the calculations would still hold.

Some important tests to assure ourselves that this procedure is correct are required. We can check we are performing a complete decomposition $\sum_{E,S} p_E^S = 1$, and make sure we are not repeating any error, $\langle \Phi_E^S | \Phi_{E'}^{S'} \rangle = \delta_{EE'} \delta_{SS'}$. Or mathematically speaking, we need to show that the set of vectors $\{|\Phi_E^S\rangle\}$ corresponds to a basis, that they are indeed orthonormal and form the complete Hilbert space. For this we can see that we only need to prove that the states $\{|\Phi_E^P\rangle\}$, $P \in \{\text{even}, \text{odd}\}$ are a basis.

We start by observing that the states $\{|\Phi_E^P\rangle\}$, $P \in \{\text{even, odd}\}$ form an orthonormal basis. Since we are considering qubits, the maximally entangled state is, up to a permutation on the qubits, a tensor product of d Bell states, as $|\Phi\rangle \equiv \bigotimes^d |\phi^+\rangle$. Then :

$$\begin{aligned} \left(\bigotimes_1^d Z \otimes \mathbb{I}\right) |\Phi\rangle &= \bigotimes_1^d |\phi^-\rangle. \\ \Pi_{(\text{even, odd})} |\Phi\rangle &= \frac{1}{\sqrt{2}} \left[\bigotimes_1^d |\phi^+\rangle \pm \bigotimes_1^d |\phi^-\rangle \right] \end{aligned}$$

And indeed,

$$|\Phi_E^P\rangle = \frac{1}{\sqrt{2}} E \left[\bigotimes_1^d |\phi^+\rangle \pm \bigotimes_1^d |\phi^-\rangle \right]$$

Then is straight forward to show that $\langle \Phi_E^P | \Phi_{E'}^{P'} \rangle = \delta_{EE'} \delta_{PP'}$ by using the orthogonality of the Bell states. The condition that $\sum_{E,P} |\Phi_E^P\rangle \langle \Phi_E^P| = \mathbb{I}$, can be derived by using that the Bell states also form a complete basis and the equality $(\bigotimes^d Z \otimes \mathbb{I}) |\phi^+\rangle = \bigotimes^d |\phi^-\rangle$ to recover the complete set of the Pauli basis.

Bibliography

- [1]Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. „Topological quantum memory“. In: *Journal of Mathematical Physics* 43.9 (2002), pp. 4452–4505 (cit. on pp. 1, 10, 13).
- [2]Andreas Reiserer, Norbert Kalb, Machiel S. Blok, et al. „Robust quantum-network memory using decoherence-protected subspaces of nuclear spins“. In: (2016). eprint: arXiv:1603.01602 (cit. on pp. 1, 19, 22, 28, 54).
- [3]H. Bernien, B. Hensen, W. Pfaff, et al. „Heralded entanglement between solid-state qubits separated by 3 meters“. In: (2012). eprint: arXiv:1212.6136 (cit. on pp. 1, 19, 24, 27).
- [4]Naomi H. Nickerson, Joseph F. Fitzsimons, and Simon C. Benjamin. „Freely Scalable Quantum Technologies using Cells of 5-to-50 Qubits with Very Lossy and Noisy Photonic Links“. In: (2014). eprint: arXiv:1406.0880 (cit. on pp. 2, 25, 29, 36).
- [5]Naomi H. Nickerson, Ying Li, and Simon C. Benjamin. „Topological quantum computing with a very noisy network and local error rates approaching one percent“. In: (2012). eprint: arXiv:1211.2217 (cit. on pp. 2, 14, 28, 29, 53, 54).
- [6]Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. New York, NY, USA: Cambridge University Press, 2011 (cit. on p. 5).
- [7]D.J. Griffiths. *Introduction to Quantum Mechanics*. Pearson international edition. Pearson Prentice Hall, 2005 (cit. on p. 5).
- [8]F. Bloch. „Nuclear Induction“. In: *Phys. Rev.* 70 (7-8 1946), pp. 460–474 (cit. on p. 5).
- [9]Martin B. Plenio and Shashank Virmani. „An Introduction to Entanglement Measures“. In: *Quantum Info. Comput.* 7.1 (Jan. 2007), pp. 1–51 (cit. on p. 6).
- [10]Denis Sych and Gerd Leuchs. „A complete basis of generalized Bell states“. In: *New Journal of Physics* 11.1 (2009), p. 013006 (cit. on p. 6).
- [11]Sergey Bravyi and Alexei Kitaev. „Universal quantum computation with ideal Clifford gates and noisy ancillas“. In: *Phys. Rev. A* 71 (2 2005), p. 022316 (cit. on p. 6).
- [12]Cédric Bény and Florian Richter. *Algebraic approach to quantum theory: a finite-dimensional guide*. 2015. eprint: arXiv:1505.03106 (cit. on p. 7).
- [13]A S Holevo and V Giovannetti. „Quantum channels and their entropic characteristics“. In: *Reports on Progress in Physics* 75.4 (2012), p. 046001 (cit. on p. 7).

- [14]M. D. Barrett, J. Chiaverini, T. Schaetz, et al. „Deterministic quantum teleportation of atomic qubits“. In: *Nature* 429 (2004), 737 EP – (cit. on p. 7).
- [15]Barbara M. Terhal. „Quantum Error Correction for Quantum Memories“. In: (2013). eprint: arXiv:1302.3428 (cit. on pp. 8, 13).
- [16]Peter W. Shor. „Scheme for reducing decoherence in quantum computer memory“. In: *Phys. Rev. A* 52 (4 1995), R2493–R2496 (cit. on p. 9).
- [17]„Multiple-particle interference and quantum error correction“. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. eprint: <http://rspa.royalsocietypublishing.org/content/452/1954/2551.full.pdf> (cit. on p. 9).
- [18]D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg. „Threshold error rates for the toric and surface codes“. In: (2009). eprint: arXiv:0905.0531 (cit. on pp. 10, 15, 34).
- [19]Bettina Heim, Krysta M. Svore, and Matthew B. Hastings. *Optimal Circuit-Level Decoding for Surface Codes*. 2016. eprint: arXiv:1609.06373 (cit. on pp. 10, 12).
- [20]Austin G. Fowler, Adam C. Whiteside, and Lloyd C. L. Hollenberg. „Towards practical classical processing for the surface code“. In: (2011). eprint: arXiv:1110.5133 (cit. on p. 12).
- [21]Sergey Bravyi, Martin Suchara, and Alexander Vargo. „Efficient Algorithms for Maximum Likelihood Decoding in the Surface Code“. In: (2014). eprint: arXiv:1405.4883 (cit. on p. 12).
- [22]Chenyang Wang, Jim Harrington, and John Preskill. „Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory“. In: *Annals of Physics* 303.1 (2003), pp. 31–58 (cit. on pp. 14, 15).
- [23]L.D. Landau and E.M. Lifshitz. *Statistical Physics*. v. 5. Elsevier Science, 2013 (cit. on p. 14).
- [24]Vladimir Kolmogorov. „Blossom V: a new implementation of a minimum cost perfect matching algorithm“. In: *Mathematical Programming Computation* 1.1 (2009), pp. 43–67 (cit. on p. 14).
- [25]Eduardo Villaseñor Alvarez. *Fault tolerance*. https://github.com/evalvarez12/fault_tolerance. 2017 (cit. on p. 14).
- [26]Wassily Hoeffding. „Probability Inequalities for Sums of Bounded Random Variables“. In: *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500830> (cit. on p. 15).
- [27]Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen. „Building logical qubits in a superconducting quantum computing system“. In: *npj Quantum Information* 3.1 (2017), p. 2 (cit. on p. 19).
- [28]Philipp Schindler, Daniel Nigg, Thomas Monz, et al. „A quantum information processor with trapped ions“. In: (2013). eprint: arXiv:1308.3096 (cit. on p. 19).
- [29]Lilian Childress and Ronald Hanson. „Diamond NV centers for quantum computing and quantum networks“. In: *MRS Bulletin* 38.2 (2013), 134–138 (cit. on p. 19).
- [30]J. Cramer. „Quantum error correction with spins in diamond“. PhD thesis. QuTech, 2016 (cit. on p. 20).
- [31]L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller. „Long-distance quantum communication with atomic ensembles and linear optics“. In: *Nature* 414 (2001). Article, 413 EP – (cit. on p. 24).

- [32]C. K. Hong, Z. Y. Ou, and L. Mandel. „Measurement of subpicosecond time intervals between two photons by interference“. In: *Phys. Rev. Lett.* 59 (18 1987), pp. 2044–2046 (cit. on p. 24).
- [33]Sean D. Barrett and Pieter Kok. „Efficient high-fidelity quantum computation using matter qubits and linear optics“. In: *Phys. Rev. A* 71 (6 2005), p. 060310 (cit. on p. 26).
- [34]F Rozpędek, K Goodenough, J Ribeiro, et al. „Parameter regimes for a single sequential quantum repeater“. In: *Quantum Science and Technology* 3.3 (2018), p. 034002 (cit. on p. 27).
- [35]David S. Wang, Austin G. Fowler, and Lloyd C. L. Hollenberg. „Surface code quantum computing with error rates over 1 percent“. In: *Phys. Rev. A* 83 (2 2011), p. 020302 (cit. on pp. 34, 46).
- [36]Ashley M. Stephens. „Fault-tolerant thresholds for quantum error correction with the surface code“. In: (2013). eprint: arXiv:1311.5003 (cit. on p. 34).
- [37]W. Dür and H. J. Briegel. „Entanglement purification and quantum error correction“. In: (2007). eprint: arXiv:0705.4165 (cit. on p. 35).
- [38]Michael Herold, Michael J Kastoryano, Earl T Campbell, and Jens Eisert. „Cellular automaton decoders of topological quantum memories in the fault tolerant setting“. In: *New Journal of Physics* 19.6 (2017), p. 063012 (cit. on p. 42).
- [39]John Watrous. *Notes for Theory of Quantum Information*. Institute for Quantum Computing. University of Waterloo, July 2011 (cit. on p. 59).
- [40]Charles H. Baldwin, Amir Kalev, and Ivan H. Deutsch. „Quantum process tomography of unitary and near-unitary maps“. In: (2014). eprint: arXiv:1404.2877 (cit. on p. 60).

