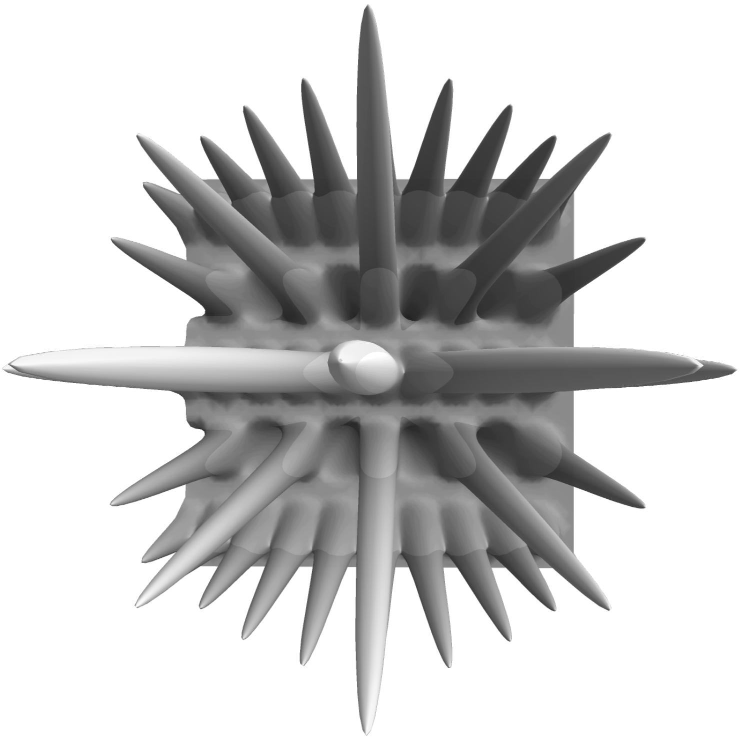

Towards fault-tolerant quantum error correction with the surface-GKP code

An extensive exploration and evaluation of the rotated surface code in concatenation with the Gottesman-Kitaev-Preskill code.

Thesis for Erasmus Mundus Master of Science in Nanoscience and Nanotechnology

Thomas Jaeken
Promotor: Giulia Ferrini
Co-promotor: Bart Sorée

Department of Microtechnology and Nanoscience
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2022

Master’s thesis 2022

Towards fault-tolerant quantum error correction with the surface-GKP code

An extensive exploration and evaluation of the rotated surface code in concatenation with the Gottesman-Kitaev-Preskill code.

Thomas Jaeken

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

KU LEUVEN

Department of Microtechnology and Nanoscience

Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2022

Towards fault tolerant quantum error
correction with the surface-GKP code
An extensive exploration and evaluation of the rotated
surface code in concatenation with the Gottesman-Kitaev-Preskill code.
Thomas Jaeken

Supervisor: Timo Hillmann, Department of Microtechnology and Nanoscience
Examiner: Giulia Ferrini, Department of Microtechnology and Nanoscience
Co-promotor: Bart Sorée, Departement Elektrotechniek

Master's Thesis 2022
Department of Microtechnology and Nanoscience
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2022

Abstract

Quantum computers have been predicted to be of great importance in the future. However, realization of this technology comes with many challenges. The fragile nature of quantum phenomena necessitates the development of fault-tolerant computation. The need for robust error correction schemes is evident. One of the most promising efforts at this time is the surface code. Recently, it became apparent that the surface code can synergize with the Gottesman-Kitaev-Preskill (GKP) code. This thesis explores that concatenated code within a circuit-level noise model approximating reality as close as possible, through classical Monte Carlo simulations relying on the state-twirling approximation and relates. We reproduce the results of ref. [1, Noh and Chamberland, Phys. Rev. A **101**, 012316 (2020)] and expand on them.

We simulate the concatenated code in different experimental setups within the parameter space of the noise model and expose relations between the results. This leads to, among others, an analogy of error-flow with current in electrical circuits. This behavior is not directly recognized in analogous simulations of the discrete surface code and was not reported previously. It corroborates the recent theory by ref. [2, Conrad et al., Quantum **6**, 648 (2022)] that the concatenation of GKP codes with stabilizer codes are a particular case of general multi-mode GKP codes. From individual simulations of the threshold for each noise source in the model, we learn that two-qubit gate noise is the most critical, while measurement noise is the most tolerable. Finally, we investigate the threshold of the concatenated code, σ^*_{gkp} as a function of the measurement efficiency and confirm the concern that this is a critical issue for practical realizations. The result of this work is a better understanding of the effect that different kinds of noise have on the logical error rate and can potentially support experimental implementations in the future.

Keywords: Quantum error correction, GKP, surface code, continuous variable quantum computing, stabilizer code, Fault-tolerance threshold, Gottesman-Kitaev-Preskill.

Acknowledgements

Above all, I would like to express appreciation to my supervisor Timo HIllmann for his belief in me. His patience and willingness to undertake extensive discussions have helped me both in achieving results and in finding confidence and joy in the work. In a broader scope, all my professors and teaching assistants at Chalmers, who have showed me how stimulating education can be. In specific I would like to acknowledge Giulia Ferrini, who has motivated and helped me both with interests this past year and in the future. Also the many professors that inspired me throughout my four years at KULeuven deserve credit for my achievements. An important part of my past year have been the friends I made in Gothenburg, who indirectly motivated me to keep going when it got hard. I want to thank explicitly my girlfriend, whose love manifested in giving me the freedom to pursue my work, while being there every time I needed a shoulder to lean on. I finally thank my family for supporting and loving me unconditionally, and always creating incredible opportunities for me that I try to do justice.

Thomas Jaeken, Gothenburg, June 2022

Contents

List of Figures

List of Tables

1

Introduction

In the last decades, the need for fault-tolerant error correction in quantum computation [3] and communication has become exceedingly evident, in order for them to realize their theoretical potential [4]. Though several physical platforms for quantum information, like solid state [5] and superconducting circuits [6], have made individual strides towards improving their reliability and scalability, each of them will clearly, at some point, need error correction to keep improving. Drawing inspiration from classical error-correction for ways to encode information redundantly and robustly, various schemes have been developed for the aforementioned platforms [7, 8, 9, 10, 11, 12] to actively supress errors, but the competition to realize fault-tolerance is still ongoing. A breakthrough event was the development of stabilizer codes by Gottesman [13].

In recent years, bosonic quantum error correction schemes have gained popularity, as they offer an intrinsic and thus hardware efficient source of redundancy for encoding. This is because the Hilbert space of a harmonic oscillator is infinite-dimensional. Several distinct ways have been proposed to encode a qubit into a harmonic oscillator, such as the cat code [14], the binomial code [15] and the Gottesman-Kitaev-Preskill (GKP) code [16]. All of them have been realized in experimental setups [17, 18, 19], and therefore their use in error-correction codes is realistic, meaningful and worth examining, as we do here.

This thesis studies specifically the GKP code, which is promising because it is robust against small displacement errors, but it has an intrinsic limit. Because of its periodic nature, any displacement larger than half the period will result in an irrevocable error. An idea introduced by ref. [20] is to concatenate quantum codes, to let them improve each other. This way, the combined code could transcend the intrinsic limit of the single-mode GKP code. Another advantage of concatenating codes is relieving the quickly growing number of necessary steps to perform an error correction cycle [13]. When an unconcatenated code is scaled up, the time it takes to perform error correction grows and may render the code useless.

Concatenating codes breaks the error correction procedure time up and avoids this problem. Several types of topological codes have been considered for this purpose [21, 22, 2, 23, 1], and we make an overview of some important results in section 3.4. The code that we study in detail is the rotated surface code, a low-density-parity-check (LDPC) code which is suitable to planar architectures. The method used for implementing this quantum circuit is gate-based quantum computing. Here, fresh qubits can be brought into the computation at any point. This stands in contrast to measurement-based quantum computing. There, one initializes all the involved qubits in a cluster state which eliminates the need for two-qubit gates.

An essential component determining the performance of the code is the classical decoder used to process the syndrome measurements. We study the efficient [24] minimum-weight perfect matching algorithm [25, 26, 27]. The efficiency directly affects the error-correction performance, because during decoding the system is unprotected from noise. Reliable estimates of this performance are obtainable through classical Monte Carlo simulations based on the state-twirling approximation [28, 29].

This thesis takes most inspiration from the original work in ref. [1]. They, however, only explore a minimal part of the parameter space of the noise model. We extend this exploration to compare noise sources and study their combined effect on the logical error-rate. We find that the independent noise sources do not interfere, and contribute independently to the surface code error rate. This is an important result and there is no clear analogy to the discrete variable (DV) surface code.

The organization of this thesis is straightforward. First, chapter 2 will treat the theoretical background of both the GKP code and the surface code and also their concatenation. After that, chapter 3 will present the main results from the work. First a section is devoted to the technicalities in quantifying figures of merit. We then report a reproduction of the results from ref. [1] and make a comparison of the rotated surface with other stabilizer codes as a concatenation partner for the GKP code. Arguably the most important important result is the simulation of a constant error-rate surface, which allows to compare the concatenated code qualitatively to the DV surface code. It also allows to formulate an analogy of the error-correction scheme with an electrical circuit. As a last study, we describe the effect of measurement efficiency on the performance of the concatenated code.

Finally, chapter 4 will provide a summary and an outlook on possible work for future projects.

2

Theory

This chapter reviews the fundamental concepts on which this thesis relies. An overview of topological and bosonic error correction is given, assuming only basic knowledge of quantum mechanics. After that, we treat the theory behind the specific focus of the thesis, the surface-GKP code. The details (pseudocode) of the simulation can be found in the appendices of ref. [1] and will not be recited here, apart from some essential pieces.

2.1 Quantum Error correction

Classical information theory is a rich field and has innumerable influences on quantum error correction. Some of those are: the idea of Hamming distance [30] or just distance, that of low-density-parity-check codes [31, 32] and the Shannon channel theory [33]. In conventional classical circuits, explicit error correction is unnecessary since the components, mainly transistors, have incredible fidelity only through digitization. However, in certain situations with increased requirements, such as extraterrestrial applications [34] or high-performance computing [35], error correction is used gratefully. In conventional communication applications, like telephone lines [36] or 4G [37], the concept of error correction is commonplace. Note that there is an essential difference with error detection, where one can simply assert whether or not an error has occurred, but instead of reverting it, discard the computation and start again.

The most fundamental principle of error correction is redundant encoding. This means we use more resources than strictly needed to represent information so that the system can tolerate a certain amount of noise without compromising the encoded information. In an algebraic sense, one should see the information as living in a sub-space, the so-called code space, embedded in the space spanned by all degrees of freedom in the physical system. The simplest way to realize redundant encoding in a classical setting is by copying the information and storing

several instances. If we assume the individual instances have a certain rate of failure, we can consider how the rate of failure of the redundantly encoded information behaves. The point where the encoding starts improving the failure rate compared to an individual bit is called the pseudo-threshold. This corresponds to a certain error rate of the individual instances and is a crucial piece of information for physical implementations. The original theory of classical fault-tolerant computing was developed by John von Neumann [38]. For scalable codes, when applying the code to a system under their threshold, we can reduce the error rate arbitrarily by increasing the code size. The threshold here is the physical error rate at which scaling up the code improves the error rate. We want a code that has a high threshold since this relieves pressure on the physical implementation.
In quantum information theory, there are some obstacles to error correction when we start from the classical techniques. What long seemed the greatest challenge is the no-cloning theorem [39] which makes it impossible to store independent copies of arbitrary states. An elegant solution for this exists in using entangled qubits instead of copies [3].

Another crucial aspect is the conservation of superpositions. We need to be able to restore a compromised state to the code space without destroying superpositions. The concept of Quantum Non-Demolition (QND) stabilizer measurements, discussed in subsection 2.1.3, realizes this.

An intuitive way to view quantum error correction is the following. The classical transistor is highly redundant because it uses millions of electrons. However, this level of coupling to the environment destroys quantum coherence. The other extreme is an unencoded physical qubit, like a single electron spin, where no redundancy is present, but the quantum behavior is apparent. We go in between these two extremes by doing quantum error correction, where we have a workable fidelity but are still in the quantum regime.

2.1.1 Quantum errors

A consequence of the fundamentally different nature of qubits compared to classical bits, is that there are many more possible errors. We assume that errors on individual qubits are independent of each other. A coherent error can be regarded as the uncontrolled application of a quantum gate. The group of quantum gates (unitary operations) on a single qubit is spanned by the Pauli matrices:

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \hat{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \hat{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.1}$$

The linearity of quantum mechanics allows us to represent a qubit's state as a 2-vector with complex entries, α and β. These entries represent the coefficients in the expansion of the state as a superposition of a pair of arbitrarily chosen orthonormal basis states. Traditionally, these basis states are the ground and excited state, $|0\rangle$ and $|1\rangle$ respectively.

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{2.2}$$

The effect of the Pauli matrices on a state is calculated by matrix multiplication as follows:

$$\begin{gathered} \hat{X}\,|0\rangle = |1\rangle\,, \hat{X}\,|1\rangle = |0\rangle\,, \\ \hat{Z}\,|0\rangle = |0\rangle\,, \hat{Z}\,|1\rangle = -\,|1\rangle\,, \\ \hat{Y}\,|0\rangle = i\,|1\rangle\,, \hat{Y}\,|1\rangle = -i\,|0\rangle\,, \\ \hat{I}\,|0\rangle = |0\rangle\,, \hat{I}\,|1\rangle = |1\rangle\,. \end{gathered} \tag{2.3}$$

Since any gate can be expanded in terms of Pauli operations, if we can correct all the separate Pauli errors (accidental applications of a Pauli operation), we can correct any coherent error. This is because during the correction procedure, the state will collapse to one of the eigenstates of the stabilizers that are measured, as will be explained in section 2.1.3. In other words, if the stabilizers are expressed in terms of Pauli matrices, the unknown gate will collapse to a Pauli operation, which we in turn correct. In practice, we only need to be able to correct the $\hat{X}$ (bit-flip error) and $\hat{Z}$ (phase-flip error) operations, since $\hat{Y} = i\hat{X}\hat{Z}$, and $\hat{I}$ is not an error.

Another error mode is the erasure error. This is an unknown Pauli-error, with the knowledge that it occurred. Essentially, if a physical qubit undergoes an erasure error, we know that we have to discard it. This is slightly less detrimental, or easier to correct, than unknown Pauli errors, because the knowledge that the qubit's state is compromised prevents us from mistaking it for a different state. We don't consider these errors further.

In reality, many errors happen because of decoherence, or coupling with an unknown quantum state in the environment. When inspecting the qubit independently from the entangled environment, it will appear mixed. For mixed states, the vector representation is insufficient. Instead, we describe states more generally using the density matrix, defined as a statistical ensemble of pure states [3].

$$\rho = \sum_i p_i \, |\psi_i\rangle \, \langle\psi_i|\,, \tag{2.4}$$

where $\sum_i p_i = 1$ and p_i is the fraction of $|\psi_i\rangle$ in the ensemble.
For any density matrix Tr$[\rho] = 1$, where the trace of a matrix is defined as Tr$[\rho] = \sum_i \langle i|\,\rho\,|i\rangle$. The purity information is retained in the coherence terms, the off-diagonals of ρ. For pure states Tr$[\rho^2] = 1$. Since we don't have knowledge about the quantum state of the environment, we have to trace it out, leaving the qubit in a mixed state.

A very important error mode is relaxation, the spontaneous tendency of qubits to evolve towards the ground state by emitting a quantum to an environment at zero temperature. A model for relaxation is amplitude damping [3]. The Kraus map for this model, with decay probability γ, is:

$$\mathcal{N}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger, \tag{2.5}$$

with Kraus operators:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \tag{2.6}$$

The Kraus operators represent the possible events that may happen to the system undergoing the channel. Either a quantum is lost, corresponding to K_1, or not, corresponding to K_0. The process of relaxation is benchmarked in terms of a figure T_1, which is the time constant in the exponential distribution of the probability that a relaxation event occured, $\gamma = 1 - e^{\frac{-t}{T_1}}$. When $\gamma = 1$, a general density matrix relaxes as follows:

$$\mathcal{N}\left(\begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix}\right) = \begin{pmatrix} \rho_{11} + \rho_{22} & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.7}$$

One can interpret this as a collapse of the Hilbert space. The two linearly independent spanning states have become dependent.

There are many other possibilities for a pure state to become entangled with its environment. An important issue is that a qubit can dephase, meaning we lose information on its phase. Dephasing can for example be modeled by considering an environment of 1 qubit, as in fig. 2.1. We illustrate the event of full depolarization, when the environment qubit is in the $|+\rangle$ state. When tracing it out, our qubit has lost all phase information, but it retains the distinction on the diagonal, as seen in eq. (2.10).

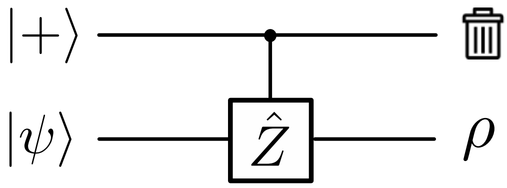

Figure 2.1: Circuit diagram representing the environment in the $|+\rangle$ state, entangled with the qubit of interest. After tracing out the environment, signified by the trash can symbol,we are left with a density matrix description of the dephased qubit ρ.

More generally, the effect of this channel can also be formulated as a linear map, with decay probability λ [3]:

$$\mathcal{N}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger, \tag{2.8}$$

with Kraus operators:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \quad K_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}. \tag{2.9}$$

When $\lambda = 1$, corresponding to the environment as in fig 2.1, a general density matrix dephases as follows:

$$\mathcal{N}\left(\begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix}\right) = \begin{pmatrix} \rho_{11} & 0 \\ 0 & \rho_{22} \end{pmatrix}. \tag{2.10}$$

A more realistic model is the phase-damping channel [3], in which the off-diagonals in the density operator decay with a time coefficient T_2:

$$\rho = \begin{pmatrix} |a|^2 & ab^* e^{-\frac{t}{T_2}} \\ a^* b e^{-\frac{t}{T_2}} & |b|^2 \end{pmatrix}. \tag{2.11}$$

Another way a qubit can decohere is depolarization, in which the qubit risks being brought to the maximally mixed state, $|\psi\rangle = I/2$, losing all information. We model this with a two-qubit environment as in fig. 2.2. We study again the event of full depolarization, when the environment is in the $|{+}{+}\rangle$ state. When we trace it out, the qubit state is maximally mixed, as in eq. (2.13).

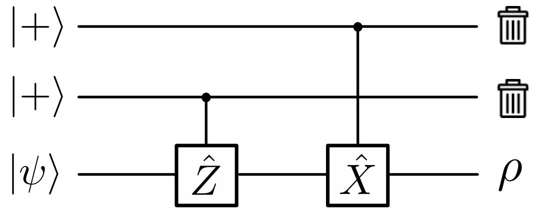

Figure 2.2: Circuit diagram representing the environment in the $|{+}{+}\rangle$ state, entangled with the qubit of interest. After tracing out the environment, signified by the trash can symbol, we are left with a density matrix description of the depolarized qubit ρ.

More generally, we can model this with a depolarization channel [3]. With decay probability p, this map acts as:

$$\begin{aligned}\mathcal{N}(\rho) &= p\frac{I}{2} + (1-p)\rho \\ &= \frac{4-3p}{4}\hat{I}\rho\hat{I} + \frac{p}{4}\left(\hat{Z}\rho\hat{Z} + \hat{X}\rho\hat{X} + \hat{Y}\rho\hat{Y}\right).\end{aligned} \tag{2.12}$$

When $p = 1$, corresponding to the environment as in fig 2.2, a general density matrix depolarizes as follows:

$$\mathcal{N}\left(\begin{pmatrix}\rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22}\end{pmatrix}\right) = \begin{pmatrix}\frac{\rho_{11}+\rho_{22}}{2} & 0 \\ 0 & \frac{\rho_{11}+\rho_{22}}{2}\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & 0 \\ 0 & \frac{1}{2}\end{pmatrix}, \tag{2.13}$$

where the last step hold because of normalization ($Tr[\rho] = 1$).

Note that if an error-correction code can correct any single-qubit error from the group spanned by {$\hat{X}_i$,$\hat{Z}_i$}, it can correct any possible error, including all decoherence errors such as depolarization or relaxation.

Note that the noise model considered in this work is different from the ones discussed here because we don't consider natural qubits but harmonic oscillators. However, this highlights the basic concepts and the principle that the ability to correct coherent errors allows to correct incoherent errors too. In this work, we will consider Gaussian displacement noise, which is decoherent noise.

2.1.2 Three qubit code

The best way to build an intuitive understanding of quantum error correction is to study a simple example. We now go through the three-qubit bit-flip code. In fig. 2.3, the circuit diagram is shown.

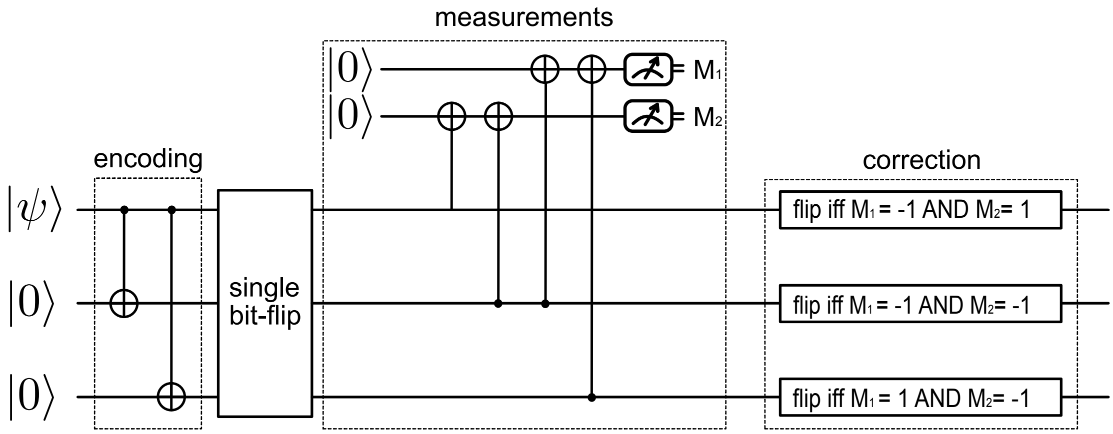

Figure 2.3: Circuit diagram of the three-qubit bit-flip code. First, the qubit is redundantly encoded in three physical qubits by entangling them with CNOT gates. Then, a potential bit-flip error occurs. Two extra ancillary qubits are invoked in the measurement stage to perform parity checks. Finally, a correcting bit-flip on one of the physical qubits can be performed based on the measurement output.

It all starts with encoding information. An arbitrary state,

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle , \tag{2.14}$$

is encoded using 3 physical qubits as:

$$|\psi\rangle = \alpha |000\rangle + \beta |111\rangle , \tag{2.15}$$

by application of CNOT gates. Note that this is entangling them instead of copying them (which would look like $|\psi\rangle = (\alpha |0\rangle + \beta |1\rangle) \otimes (\alpha |0\rangle + \beta |1\rangle) \otimes (\alpha |0\rangle + \beta |1\rangle)$). When an erroneous bit-flip happens on an arbitrary physical qubit, for instance, the first qubit:

$$|\psi\rangle = \alpha |100\rangle + \beta |011\rangle , \tag{2.16}$$

we can detect this by measuring the observables $\hat{Z}_1\hat{Z}_2$ and $\hat{Z}_2\hat{Z}_3$, called the stabilizers. To measure these without decoding the information, we entangle one ancillary for each. The reason measuring these observables does not collapse the logical qubit, is because they commute with the logical operators, $\hat{X}_L = \hat{X}_1\hat{X}_2\hat{X}_3$,$\hat{Z}_L = \hat{Z}_i$ where $i = 1, 2, 3$. In other words, the stabilizers have no impact on the code states: $\hat{Z}_1\hat{Z}_2 |\psi\rangle = |\psi\rangle = \hat{Z}_2\hat{Z}_3 |\psi\rangle$. The outcome lets us identify which physical qubit has undergone the bit-flip. For example, if the outcome of $\hat{Z}_1\hat{Z}_2$ is $+1$, we know that the two first physical qubits are in the same state, but not which state this is. If the outcome would be -1, we would know that those two qubits are in a different state. This means one of them has undergone a bitflip. Combining this with the outcome of $\hat{Z}_2\hat{Z}_3$, we know which qubit has undergone an error. Therefore we are able to revert this error, preserving the logical superposition. However, when two bit-flip errors would happen simultaneously, the code would conclude that the initial state was the opposite, resulting in a logical error. The three-qubit bit-flip code can correct only single-qubit bit-flip errors and cannot correct phase-flip errors. The pseudo-threshold of this code can be found as the solution of:

$$p = p^3 + 3p^2(1-p) \to p_{th} = 1/2 \tag{2.17}$$

Where p is the logical bit-flip error rate of a bare qubit, and $p^3 + 3p^2(1-p)$ that of a logical qubit encoded with the three-qubit bit-flip code. As soon as the bit-flip probability $p > 1/2$, applying the code improves the performance.

This code is part of a class of codes called stabilizer codes, which section 2.1.3 will treat more generally.

2.1.3 Stabilizer codes

The idea of stabilizer codes [13] is that we can do syndrome measurements that tell us in which way the state has left the code space without telling us anything about where in the code space the state would be. Thus not collapsing any superposition the logical state may be in but giving us the information needed to restore the state to the code space. In a physical system of n qubits, one can encode k logical qubits if $k \leq n$. In this work, we only consider a single logical qubit. Measuring stabilizers collapses the state into one of their eigenspaces. Each eigenspace is at least two-dimensional such that the logical superposition is preserved within. If they are more than two-dimensional, the code is called degenerate. This is a consequence of the careful choice of stabilizers. Whenever the measurement outcome is different from the set of eigenvalues corresponding to the code space, it tells us how the state is compromised so we may restore it. One always measures one degree of freedom less than are in the physical Hilbert space, leaving one to keep the logical qubit intact.

From a more rigorous and mathematical point of view, we can say the following, denoting noise events as operators E_i [13, 22]:

The code space is defined as the common +1-eigenspace of an abelian[a] subgroup of the physical Hilbert space, $\mathcal{S}$, called the stabilizer group. The stabilizer operators commute with all logical operators. All operators that are not logical are

---

[a] An Abelian, or commutative, group is a mathematical group where the order of 2 applied group operations does not affect the result.

errors and don't commute with the stabilizers, allowing us to detect them. All this can be summarized in the necessary and sufficient condition, found by ref. [40] and [41]:

$$\langle\psi_i| E_a^\dagger E_b |\psi_j\rangle = C_{ab}\delta_{ij}, \tag{2.18}$$

where E_a and E_b are all the errors the code can correct, and $|\psi_i\rangle$ and $|\psi_j\rangle$ are all code states. A special property of stabilizer codes is that all operators, including error operators, when multiplied with any stabilizer, are left equivalent and with the same stabilizer measurement syndrome. If any stabilizer, acting on a logical basis state, leaves the state orthogonal, the code is called degenerate. This is equivalent to saying that different elements of $\{E_i\}$ don't all produce linearly independent states [42]. Therefore, when a quantum code is degenerate, some errors are indistinguishable from each other by stabilizer measurements, yet they are all correctable. Since they are indistinguishable, the applied correction is identical. This is why the physical space spanning the logical states needs to be more than two-dimensional.

Quantum codes can also be non-degenerate, when all distinct errors in $\{E_i\}$ produce linearly independent states. For these codes, the quantum Hamming bound [3] applies. It formalizes that the error space should fit in the stabilizer space:

$$\sum_{j=0}^{t} \binom{n}{j} m^j 2^k \leq 2^n, \tag{2.19}$$

where we encode k logical qubits into n physical ones, and up to t errors of m kinds can be corrected. If the bound is saturated, the code is called perfect.

An extension of the degeneracy concept is purity. A code said to be pure to t if and only if the stabilizer group has no elements with weight less than t [43], except the identity. If a code is pure to its distance, we say the quantum code is pure. Equivalently, for a pure code, all distinct errors in $\{E_i\}$ produce orthogonal states. The three-qubit code is non-degenerate and pure, with $m = 1, t = 1$ for only single-qubit bit-flip errors. The quantum Hamming bound evaluates to: $2 + 3 * 2 = 8$, so it is a perfect code with regard to the error set $\{\hat{X}_1, \hat{X}_2, \hat{X}_3\}$. The rotated surface code, the focus of this thesis, is degenerate. The performance of degenerate codes is better than that of non-degenerate codes [44].

The distance of a stabilizer code is the weight of the minimal logical operator [3]. Here, the weight signifies the amount of physical qubits that are affected by the operator. In the three-qubit bit-flip code, the minimal logical operator is $\hat{Z}_i$, where $i = 1, 2, 3$. Therefore, the distance of that code is 1. More formally, the distance is the weight of smallest $E = E_a^\dagger E_b$ for which eq. (2.18) doesn't hold. A code of distance $d = 2t + 1$ can correct at most t errors, where t is called the correction threshold, the weight of the maximal error operator past which it can no longer be corrected. For $\hat{X}_i$ errors in the three-qubit bit-flip code, the correction threshold is 1. For $\hat{Z}$ errors, it is 0, meaning no phase flip can be corrected.

Another way to formalize the conditions for error-correction on stabilizer codes is the theorem 10.8 from ref. [3]. It tells us that a code with stabilizer group S can correct all errors that satisfy $E_j^\dagger E_k \notin N(S) - S \forall j, k$. $N(S)$ is the normalizer of S, containing all elements A such that A commutes with all elements in S. The code fails when $E_j^\dagger E_k$ is a logical operation. This integrates the correction threshold. If we consider, for example, the three-qubit bit-flip code and an error set $\{\hat{I}, \hat{X}_1, \hat{X}_2, \hat{X}_3, \hat{X}_1\hat{X}_2\}$ that contains a two-qubit bit-flip error, the theorem tells us that the code cannot correct this set of errors. Because for $E_j^\dagger E_k = \hat{X}_3^\dagger \hat{X}_1 \hat{X}_2 \in N(S) - S$, the theorem is violated.

A subclass of the stabilizer codes is that of Calderbank-Shor-Steane (CSS) [45, 46] codes. These are codes where the each stabilizer contains either only $\hat{Z}_i$ or $\hat{X}_i$ on the individual physical qubits. The stabilizers descend from parity check codes of classical codes. This is one reason by CSS codes are of interest. They allow simple translation of intensely explored classical codes. Additionally, they allow transversal fault-tolerant multi-logical-qubit interactions. Transversality means we can treat all physical qubits in a logical qubit equally. In this thesis we don't consider interactions between logical qubits.

The rotated surface code that is the topic of this thesis is a CSS code. Because of the restriction on the stabilizers, their performance cannot be as good as the optimal general code, but they still hold potential to be sufficient. [13]

2.2 Topological codes

Information can be encoded robustly as topological properties of a surface. This is an inherently scalable strategy. Here, we present it using discrete variable (DV), unencoded qubits as the building blocks. The first example of a topological code was the toric code [47, 48, 49], but many variations exist at this time [50, 51, 52]. To have an intuitive sense of how this works, imagine a torus as in fig. 2.4. A torus is not a simply connected surface because we can conceive a loop on it that is not contractible without leaving the surface. This is precisely what the toric code exploits. The logical qubit is encoded in the parity of incontractible loops, which is a highly degenerate property. A logical $|1\rangle$ is encoded by an odd number of incontractible loops, while an even number of such loops represents a logical $|0\rangle$.

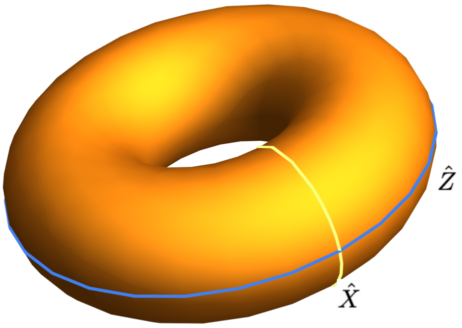

Figure 2.4: A toric surface with illustrations of a bit-flip $\hat{X}$ and a phase-flip $\hat{Z}$ operation.

Imagine a corrupted loop. As long as the interruption of the loop is not longer than half its perimeter, the "distance" of the code, the loop can be restored. In fact, the number of qubits that can be encoded on an orientable, closed surface is determined by its genus. For open surfaces, one more qubit than the number of holes can be encoded, when appropriate boundary conditions are maintained.

2.2.1 Rotated Surface code

The specific code we now study is the discrete Rotated Surface code, an example of a topological CSS [3] code. It exists in a plane, which makes it naturally easier to implement in contemporary technologies [53, 54, 55]. The surface we consider is a square grid of $d \times d$ physical qubits, and we impart some special properties to the edges. Fig. 2.5 shows a schematic overview, with a clarification in fig. 2.6. Throughout this thesis, anytime the "surface code" is referenced, this means the rotated surface code. A logical $|1\rangle$ ($|0\rangle$) is encoded by an odd (even) number of chains connecting the top and bottom edges. A logical $|+\rangle$ ($|-\rangle$) by an odd (even) number of chains connecting the left and right edges. Both logical states are degenerate and are spanned by all states generated by application of the surface code stabilizers to an arbitrary instance.

Note that d is always taken to be odd. The reason for this is that the surface code is, in a sense, the product of two repetition codes. In repetition codes, the number of qubits is always odd because the ability to correct errors does not improve when one adds one more qubit. This stems from the majority vote decoding that always needs a deciding (odd) vote. All the while, the technical requirements would be considerably increased. So clearly, even d are not beneficial.

Another argument could be made that the number of stabilizers $\frac{d^2-1}{2}$ signaling $\hat{Z}$ and $\hat{X}$ errors should be equal.

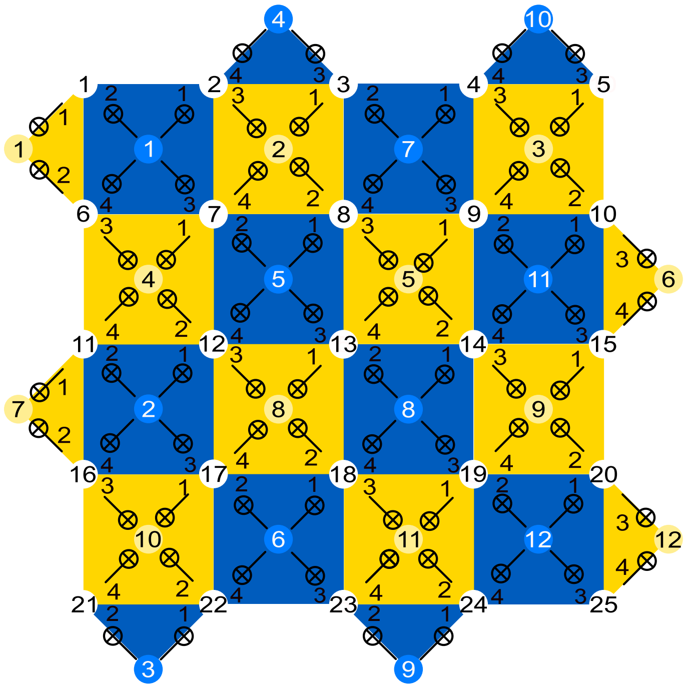

Figure 2.5: The surface code for d=5. The data qubits are in white circles. The surface code ancillaries are in blue and yellow circles for X-type and Z-type syndromes respectively.

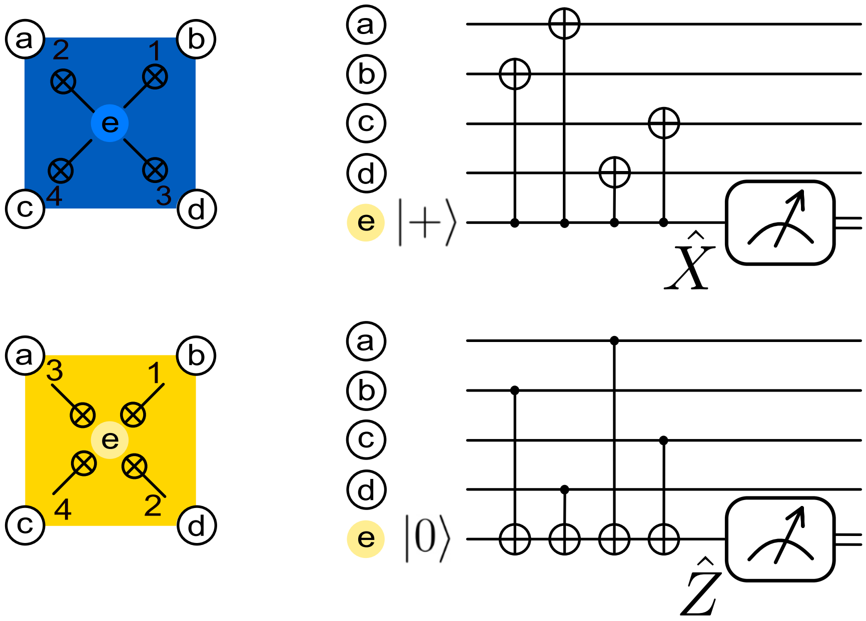

Figure 2.6: Circuit excerpts from fig. 2.5 with their translation in conventional circuit notation to show more explicitly how the notation works. The indices within the squares refer to the order in which the CNOT gates are performed.

The distance of the code is equal to the side of the square surface, d. In principle, one could conceive a rectangular code, which would protect better against either $\hat{X}$ or $\hat{Z}$ errors than against the other [56]. Two sub-lattices of ancillary qubits are intertwined with the data qubits to enforce ”rules” on the grid. Ancillas of one grid measure the stabilizers $\hat{Z}_a\hat{Z}_b\hat{Z}_c\hat{Z}_d$ of their nearest data qubits a through d. The ones of the other measure the stabilizers $\hat{Z}_c\hat{Z}_d\hat{Z}_e\hat{Z}_f$ of their nearest data qubits c through f. The fact that each ancillary qubit only interacts with its nearest neighbors makes this code an LDPC code [32]. For a general code, this means each qubit only interacts with a bounded number of generators, constant in the number of physical qubits. This is a desirable property. It improves scalability because it minimizes the required coherence time of the ancillaries. The positioning of the sublattices is defining for the ”rotated” surface code. The grid has ”rough” and ”smooth” edges pairwise on opposing sides. The ”smooth” edges are the ones that, if connected by an odd number of chains, represent a logical $|1\rangle$. It is this boundary condition that gives the, otherwise ordinary, open surface the same topological power as a closed surface of genus one. Note that any chain, contractible on the surface, is ignored by the stabilizer measurements and does not affect the logical state. Contractible means that the chain can disappear by applying only stabilizer operations like in fig. 2.7. The stabilizers are measured by locally entangling ancillary qubits with four or two data qubits and counting the parity of that local group coherently as in fig. 2.6. When this information from all groups is combined, one can reconstruct chains by connecting the syndrome ancillaries that measured odd parity. How the syndrome outcomes are processed into suspected error chains is a separate issue and is discussed in detail in section 2.2.3. We do $d^2 - 1$ surface syndrome measurements, the minimum required amount for the implementation with DV physical qubits. Later, we will consider continuous variable (CV) qubits, and section 2.4 will discuss the implications of that.

It is interesting to note that this is a degenerate code because it can correct more errors than it can uniquely identify. A simple example to illustrate this point is shown in figs. 2.8. A syndrome measurement cannot pinpoint the exact physical qubits that underwent an error, but it can still correct the logical state. To operate this qubit, coordinated control across the grid is necessary. Several successful research efforts have resulted in the formulation of appropriate schemes for this [50, 57, 58]. However, the focus of this thesis is a quantum memory, so logical operations are not further discussed.

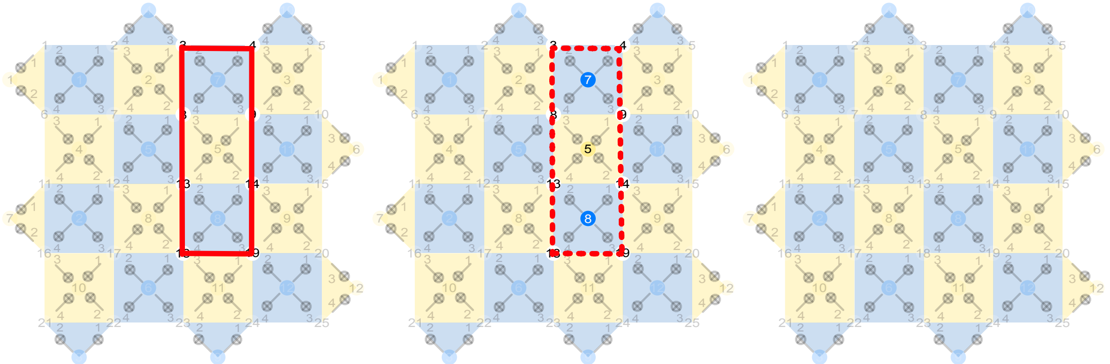

Figure 2.7: (a) Illustration of a contractible loop showing the degeneracy of the logical $|0\rangle$ state. By applying the stabilizer operator $\hat{X}_3\hat{X}_4\hat{X}_8\hat{X}_9$,$\hat{X}_{13}\hat{X}_{14}\hat{X}_{18}\hat{X}_{19}$ the chain disappears without affecting the logical state.

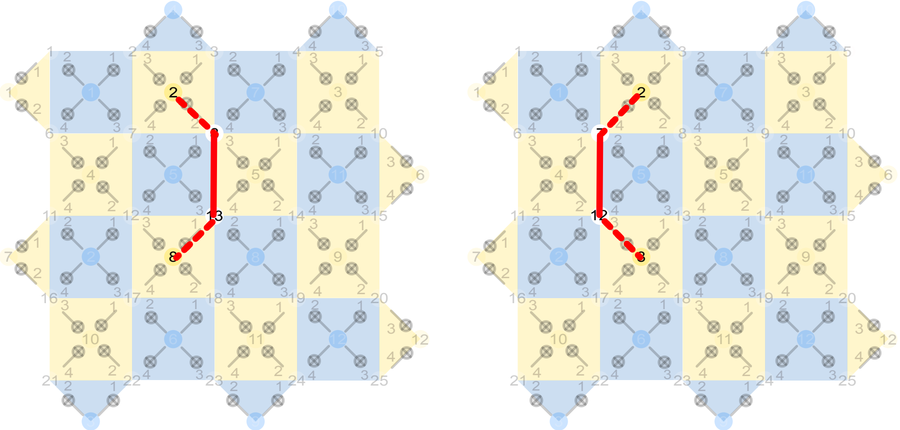

Figure 2.8: Illustration of how a certain syndrome signature cannot uniquely identify the error chain. Only syndrome qubits Z_8 and Z_2 measure odd parity, but we show two instances from the set of error chains with equal length that could cause this. Even if the decoder picks the wrong one, the error is still corrected because a contractible loop would result, which is inside the degenerate logical $|0\rangle$.

2.2.2 Noisy syndrome measurement

This thesis studies error correction codes in the framework of a specific error model. The details of this model have to wait until after the introduction of the GKP code and will be discussed in section 2.4.1. However, it needs to be clear at this stage that the error model considers faulty components. In other words, the error correction procedure itself brings in extra errors, as is realistic for any platform. To mitigate the noise in the syndrome measurement process and thus improve the performance of the code, we repeat the procedure several times.

Since logical errors scale in time, the same way they do in space [50], it is natural to repeat d times, where d is the distance of the code. We say the error correction procedure has d rounds. This means the four steps in the surface syndrome measurement in fig. 2.6 are repeated d times. One can interpret this as improving the estimate the syndrome by combining information from separate measurements.

2.2.3 Decoding

After obtaining the information from all the syndrome measurements, they need to be processed by a decoder algorithm to restore the states. Note that whenever we refer to decoding in this thesis, it means processing syndrome measurements to apply error correction, not decoding in the sense that the protected information would be returned to its bare state or be read out. This part of the simulation is not a classical approximation, as it will always be classical in physical implementations.

Several decoding strategies are known, each with its advantages and disadvantages. Two of the most studied ones are the perfect minimum weight matching (MWPM) [25, 26, 27] and the maximum likelihood decoder [59, 60, 22]. The most basic technique would be to keep a look-up table that matches syndrome signatures to strings of correction operators. This is not an optimally scalable strategy, but it may be used to complement more sophisticated methods [61]. On the other side of the spectrum, the novel technique of machine learning can be applied to decode the syndromes [62, 63, 64]. We now focus on the MWPM technique used to continue this thesis. It is an efficient algorithm, scaling as $\mathcal{O}(n)$ for serial execution, where n is the number of detection events, and as $\mathcal{O}(1)$ for parallel execution [24]. For a matching to be ”perfect” means that all vertices are incident on exactly one edge of the matching. As shown in fig. 2.9, there is not a unique matching that obeys this requirement. For it to be a ”minimum weight” matching, the sum of weights of all edges in the matching should be the minimum out of all possible perfect matchings.

As the goal is to simulate error correction codes classically, we need a way to classify a sample as being corrected or not. Therefore, after the d rounds of noisy surface stabilizer measurements, a noiseless round is executed in order to be able to judge whether the error correction procedure was successful. The edges of a graph, holding all $d \times d \times (d+1)$ syndrome qubits as vertices, keep track of the physical layout, as in fig. 2.10.

The graph consists of d+1 layers, and there are horizontal edges that connect the

vertices in all places where a data qubit is present. The weight that these edges hold is determined by their history, as will be addressed later in this section.
The d+1 layers along the time axis are then connected by edges at all syndrome qubit positions, again with appropriate weights. Along with vertices corresponding to the syndrome measurements, some extra nodes enforce the boundary conditions. These pseudo-vertices make up the ”smooth” edges of their appropriate sub-lattice, and their connecting edges have zero weight since they do not correspond to a physical qubit. This means that an error chain can continue from one ”smooth” edge to the other at no cost in terms of weight. This is where boundary conditions of the surface code differ from those of the toric code, where the boundary qubits would be paired one-to-one with those on the opposite boundary.

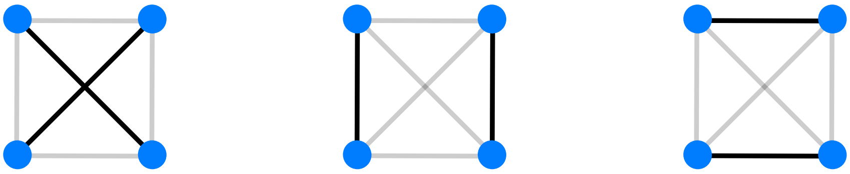

Figure 2.9: Illustration of the three different perfect matchings that can be made from four vertices in a fully connected graph.

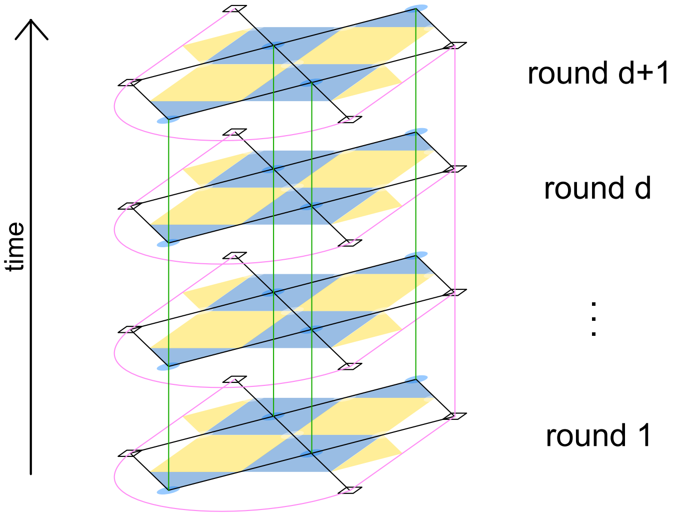

Figure 2.10: Schematic of the d+1 rounds of syndrome measurements organized in a 3D graph for d=3. The pink edges have zero weight, and the vertical green ones have weights based on the error probability of the syndrome ancillary qubits. The black edges are the horizontal ones with weights corresponding to the error probability of the data qubits.

Now that the measurement data is appropriately organized, the actual calculation can start. First, we distill a fully connected graph from the whole grid by calculating the shortest paths between all flagged syndromes. For this, Dijkstra's algorithm [65] is used in the implementation of the Networkx package [66]. It finds the shortest path from a source node to all other nodes. We now have a reduced graph containing only the flagged syndrome qubits where the weights on the edges between each pair of nodes are the shortest path between these nodes in the original graph. In order to correct the errors on the surface, we need to identify error chains. For that, we use the Networkx package [66] MWPM algorithm to pair the flagged syndrome qubits up. It implements the ”blossom” method, which finds augmenting paths, in combination with the ”primal-dual” method, which finds a maximal weight matching [67]. By giving inverse weights, it finds the minimum weight matching instead. At this point, the degeneracy of the code, explained in section 2.2.1, becomes of interest. A matching of the flagged syndrome qubits does not correspond to a unique error chain landscape. However, because of the degeneracy of the code, any minimum weight chain will correct the state back into the code space. The total weight that the decoder minimizes is the sum of the weights in the strings between the matched-up pairs. The weight of a single edge is related to the expected noise on the corresponding qubit. By carefully keeping track of how noise propagates through the grid during the four steps, one obtains for every qubit a probability that it has undergone an error based on the assumed error model. The error probability of the data qubits determines the horizontal edge weights, while that of the ancillary qubits determines the vertical ones. These uneven weights are the origin of boundary effects. Typically, the boundary syndromes undergo fewer gates and more idling, making them behave differently under the influence of noise. One consequence of this, as we will see in section 3.2, is that the error correction threshold depends on the distance of the surface code. Section 3.3 discusses the weights for the surface code concatenated with the GKP code. Since we consider a realistic error model, some physical qubits are more likely to have undergone an error based on their history and position. The data and syndrome ancillary qubits will have different expected error profiles. Even inside those groups, the interaction histories exhibit significant differences. In section 3.3 the complete set of weights is stated, which is obtained by carefully studying and keeping track of the evolution of quadrature displacements throughout the whole procedure.

2.3 GKP error correction

2.3.1 Continuous variable quantum systems

Some quantum systems naturally have only two eigenenergies, meaning their state lives in a two-dimensional Hilbert space, otherwise known as a qubit, such as the spin of an electron or the polarization of a photon [68, 69, 70, 71]. There is, however, a plethora of quantum systems that do not fit this mold but can also be used to hold quantum information. Bosonic systems are an important such group. Specifically the modes of a harmonic oscillator, such as can be constructed in superconducting [72, 73, 6] or optical [74, 75, 76, 77, 78] quantum circuits [3]. The quantum states of these systems have a continuous spectrum in the quadrature eigenvalues and can be described by an infinite Hilbert space.

Bosonic Hilbert spaces can be spanned by the 2 quadratures with continuous eigenspectrum, which is most natural in many cases:

$$[\hat{q}, \hat{p}] = i, \tag{2.20}$$

where we assumed $\hbar = 1$. The non-zero commutator means that a quantum state cannot be a point in the space spanned by $\hat{p}$ and $\hat{q}$. There needs to be sufficient uncertainty in the state. One good way to represent states in phase space is by making use of the Wigner-transformation of the density matrix, which performs the following:

$$W(q,p) \equiv \frac{1}{2\pi}\int_{-\infty}^{\infty} \langle q + \frac{1}{2}x|\, \hat{\rho}\, |q - \frac{1}{2}x\rangle\, e^{ixp} dx. \tag{2.21}$$

This is a *quasi-probability* distribution and thus some care should be taken not to interpret it as a traditional probability distribution.

There is a possible change of basis that allows spanning the infinite Hilbert space with discrete energy levels. This is called second quantization and the levels are called photons in case the system we study is an electromagnetic mode. The change of basis is as follows:

$$\begin{aligned} \hat{a} &= \frac{\hat{q} - i\hat{p}}{\sqrt{2}}, \\ \hat{a}^{\dagger} &= \frac{\hat{q} + i\hat{p}}{\sqrt{2}}, \end{aligned} \tag{2.22}$$

where $\hat{a}^{\dagger}$ has no eigenstates, and $\hat{a}$ has a continuous eigenspectrum, with eigenstates called ”coherent” states that span the entire Hilbert space. The operators

$\hat{a}^\dagger$ and $\hat{a}$ are clearly not Hermitian and therefore not observable. An operator that is observable however is their product:

$$\hat{n} = \hat{a}^\dagger \hat{a} = \frac{\hat{p}^2 + \hat{q}^2}{2}. \tag{2.23}$$

This operator counts the number of photons in a state, and the eigenvalues of this state span a Fock space. We could express coherent states in terms of the basis states of the Fock space: $|\alpha\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle$. Therefore, this basis is transformed but equivalent to the coherent state basis.

Quantum information processing utilizing the full continuous spectrum of these states is theoretically possible and has been the subject of several studies [75, 76, 77, 78]. It has, however, been proven that fault-tolerant computation is impossible with Gaussian input states [79, 80], thus it could only be used in quantum simulations where errors are less detrimental.

However, a different use for this infinite Hilbert space is as a source of large degeneracy. Principally, we define a 2D subspace in the infinite Hilbert space where the qubit lives, the code space. This allows error correction in a hardware-efficient way and lets us recycle qubit algorithms because a harmonic oscillator effectively becomes an error-resistant qubit.

Several encodings of a qubit in this system have been proposed. Each of them has its natural representation system. An example is the binomial code [15], which naturally presents itself in the Fock basis. Another example is the family of cat codes [14], which the coherent state basis suits best. The code that we study in the next section 2.3.2 is most straightforwardly described in the quadrature basis. These codes all do allow fault-tolerance [79, 16, 81, 14].

2.3.2 Ideal GKP-qubits

The codes proposed by Gottesman, Kitaev and Preskill in 2001 [16], also named GKP codes, are based on the stabilizer principle. They have been shown to have theoretical advantages over cat codes when it comes to performance [82] leading to studies into practical implementations [83, 84].

The canonical, unnormalized GKP basis states, expanded in the quadrature basis, are:

$$|0\rangle = \sum_{n=-\infty}^{\infty} |2n\lambda\sqrt{\pi}\rangle_q = \sum_{n=-\infty}^{\infty} |n\frac{\sqrt{\pi}}{\lambda}\rangle_p, \tag{2.24}$$

$$|1\rangle = \sum_{n=-\infty}^{\infty} |(2n+1)\lambda\sqrt{\pi}\rangle_q = \sum_{n=-\infty}^{\infty} (-1)^n \, |n\frac{\sqrt{\pi}}{\lambda}\rangle_p \, . \tag{2.25}$$

The parameter λ is free to be chosen, and can be optimized to the noise model. For example, when the noise in one quadrature is expected to be much more, it is better to have larger period there, as that allows to cope with larger displacements. For the continuation of this thesis, the parameter $\lambda = 1$ because we do not consider a biased noise source. These basis states are perfectly orthogonal, and any displacement error smaller than $\frac{\sqrt{\pi}}{2}$ will result in a mapped but still orthogonal version of the code space. By the Knill-Laflamme condition [40], this means we can correct these kinds of errors. It is essential to realize that an arbitrarily displaced GKP state is now outside the code space and is different from a logical superposition state. Instead a superposition state can be obtained by the logical Hadamard gate, which acts like a rotation in quadrature space. The complete logical Clifford gate-set for GKP qubits is:

$$\hat{S}_{gkp} = \text{exp}\left[i\frac{\hat{q}^2}{2}\right], \quad \hat{H}_{gkp} = \text{exp}\left[i\frac{\pi}{2}\hat{a}^\dagger\hat{a}\right], \quad CNOT_{gkp}^{j\to k} = \text{exp}\left[i\hat{q}_j\hat{p}_k\right]. \tag{2.26}$$

The stabilizer operations corresponding to the encoded states in eqs. (2.24) and (2.25) are shifts in the quadratures. Because the canonical encoding is a superposition of periodic quadrature values, a shift of one period does not affect the state.

$$\hat{S}_q = e^{i2\sqrt{\pi}\hat{q}}, \quad \hat{S}_p = e^{-i2\sqrt{\pi}\hat{p}}. \tag{2.27}$$

The logical operators on the encoded qubits are as follows:

$$\hat{Z}_{gkp} = \sqrt{\hat{S}_q} = e^{i\sqrt{\pi}\hat{q}}, \qquad \hat{X}_{gkp} = \sqrt{\hat{S}_p} = e^{i\sqrt{\pi}\hat{p}}. \tag{2.28}$$

The complementary basis states can be obtained by applying the Fourier transformation $\hat{F} = e^{\frac{i\pi}{4}(\hat{p}^2+\hat{q}^2)}$ to the basis states from eqs. (2.24) and (2.25):

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \sum_{n=-\infty}^{\infty} |n\frac{\sqrt{\pi}}{\lambda}\rangle_q = \sum_{n=-\infty}^{\infty} |2n\lambda\sqrt{\pi}\rangle_p \, , \tag{2.29}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \sum_{n=-\infty}^{\infty} (-1)^n \, |n\frac{\sqrt{\pi}}{\lambda}\rangle_q = \sum_{n=-\infty}^{\infty} |(2n+1)\lambda\sqrt{\pi}\rangle_p \, . \tag{2.30}$$

2.3.3 Resilience to noise

The advantage of the GKP encoding is that considerable noise does not necessarily destroy the logical information. Directly, the code protects the logical information from quadrature displacement events, but as we will see later, this protects

it from most kinds of errors.
Just the passive, redundant encoding alone gives us some, but very little, resilience. The real power comes from the fact that displacements can be measured and reverted up to a modular value of $\sqrt{\pi}/2$. This must and does happen without collapsing the logical qubit. We detail here the Steane type of error correction, which is the norm for gate-based implementations. We measure the stabilizer operators by entangling an ancillary GKP-qubit to the data qubit. This may be realized in a number of ways, but we use a SUM ($SUM_{j\to k} = e^{-i\hat{q_j}\hat{p_k}}$) and inverse SUM ($SUM^{\dagger}_{j\to k} = e^{i\hat{q_j}\hat{p_k}}$)) gate and perform a modular homodyne measurement of its quadrature as in fig. 2.11.

This tells us how far from the code space the state has been erroneously displaced and allows us to revert this.

Interestingly, we are able to measure both quadratures in spite of the Heisenberg principle. This is because they are both measured modulo $\sqrt{\pi}$, and therefore the observables in eq. (2.27) commute. Intuitively, the measurement outcome gives us no information on which peak the state is closest to, so we learn nothing about the encoded state. The canonical GKP states have an infinite photon number, meaning they are nonphysical. The following section will treat the physical approximation with a finite photon number.

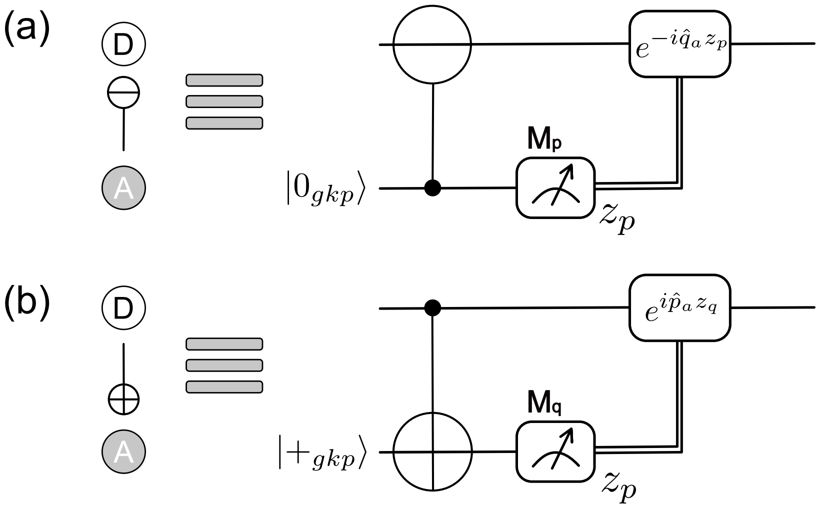

Figure 2.11: Circuit diagrams for the (a) $\hat{q}$ and (b) $\hat{p}$ GKP gadgets. First a SUM gate, followed by a homodyne measurement of the ancillary quadrature, which results in the value z_q or z_p. Finally, a conditional shift of this modular value is performed to restore the data qubit to the code space.

2.3.4 Physical GKP code states

The ideal GKP states introduced in the previous subsection are not physically realizable because they are not normalized and thus have infinite energy. Any physical implementation will have finite energy, which translates into spreading of the peaks in the quadrature space. We can model this using a coherent Gaussian envelope on an ideal GKP state: [1]

$$|\psi^{\Delta}_{gkp}\rangle \propto \int d^2\alpha e^{-\frac{|\alpha|^2}{2\sigma^2_{gkp}}} \hat{D}(\alpha) |\psi_{gkp}\rangle , \tag{2.31}$$

where $|\psi_{gkp}\rangle$ is any superposition of eq. (2.24) and (2.25). The squeezing of the GKP peaks, represented by σ_{gkp}, is balanced by the Gaussian envelope width, determined by Δ, to normalize the state. They are related as: $\sigma^2_{gkp} = (1-e^{-\Delta})/(1+e^{-\Delta}) \xrightarrow{\Delta \ll 1} \Delta/2$, as derived in ref. [1]. Refs. [85, 86, 87, 88, 89] provide more information on approximate versions of a GKP state. The Wigner representations of the GKP computational basis states after this coherent channel look like fig. 2.12. We can distinguish the periodic structure as well as the interference pattern from the coherent superposition in eq. (2.31).

The physical GKP states are no longer perfectly orthogonal. There is a finite probability of mistaking one for the other without any failure in the measurement procedure, just because the states overlap slightly. Therefore, they are not true qubit states in the sense that a true qubit lives in an orthogonal Hilbert space. Fig. 2.13 shows this error probability. It sums the integrals of the Gaussian squeezed state probability distribution over the "bins" that correspond to the $|0\rangle_{gkp}$, segments of $\sqrt{\pi}$ centered around the peaks of the ideal state in (2.24).

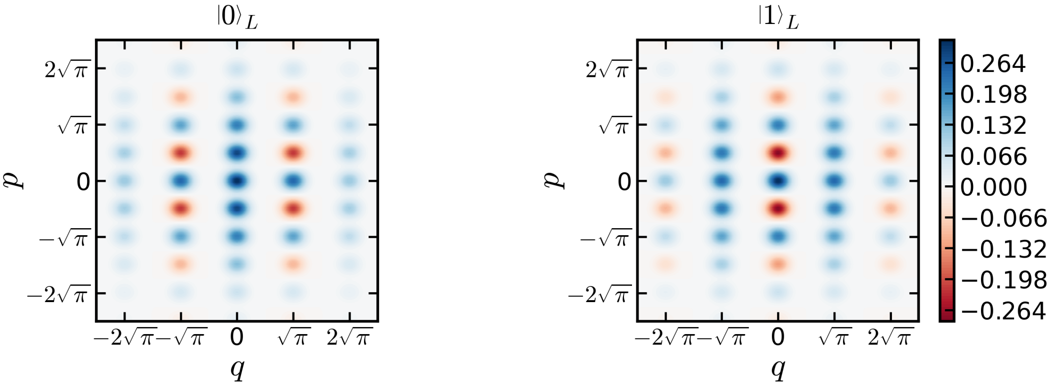

Figure 2.12: Contour plot of the Wigner function representation of the logical basis states of the GKP code. Notice that the states, apart from the envelope, are the same up to a $\sqrt{\pi}$ displacement along the q axis.

The integral of a Gaussian distribution is the error-function:

$$
\begin{aligned}
p_{err}[\sigma] &\equiv \sum_{n\in\mathbb{Z}} \frac{1}{\sqrt{2\pi\sigma^2}} \int_{(2n+\frac{1}{2})\sqrt{\pi}}^{(2n+\frac{3}{2})\sqrt{\pi}} d\xi \, \mathrm{exp}\left[-\frac{\xi^2}{2\sigma^2}\right] \\
&= \frac{1}{2}\sum_{n\in\mathbb{Z}} \mathrm{erf}\left(\sqrt{\frac{\pi}{2}}\frac{2n+3/2}{\sigma}\right) - \mathrm{erf}\left(\sqrt{\frac{\pi}{2}}\frac{2n+1/2}{\sigma}\right).
\end{aligned}
\tag{2.32}
$$

In order to be able to make a classical simulation [90], we must consider an incoherent error channel instead. Therefore, we perform a state-twirling with stabilizer operators, resulting in: [28, 91, 29]

$$
\mathcal{N}[\sigma_{gkp}](\hat{\rho}) = \int \frac{d^2\alpha}{\pi\sigma^2} e^{-\frac{|\alpha|^2}{\sigma^2}} \hat{D}(\alpha)\hat{\rho}\hat{D}^\dagger(\alpha), \tag{2.33}
$$

where $\rho = |\psi_{gkp}\rangle \langle\psi_{gkp}|$ and we integrate over the real and imaginary parts of α. This incoherent error channel leads to overestimating the error rate and gives a lower, conservative bound on the threshold. For the justification, see Appendix A in ref. [1]. Recently, it was found that this representation accurately reproduces the error rates obtained from the numerically exact state given in eq. (2.33) [92, 93, 94]. Using this will yield a good approximation of the actual threshold [95].
We show now, following ref. [1], how one arrives from the coherent superposition [eq. (2.31)] to the incoherent superposition [eq. (2.33)], by state-twirling [41] with stabilizer operators.

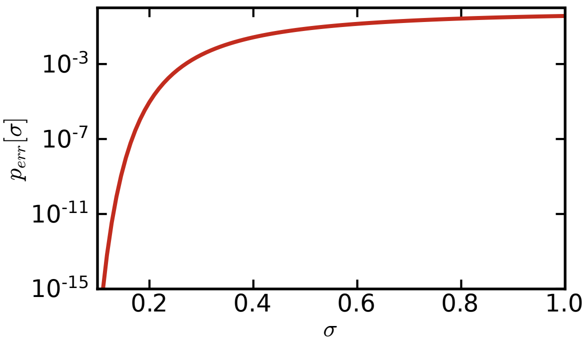

Figure 2.13: Probability of an erroneous state identification as a function of the standard deviation in quadratures of the physical state in eq. (2.31)

State-twirling means to randomly apply elements of a unitary group to an input state. Because of the properties of the group, this diagonalizes the state, making it efficiently classically simulatable. In our case, this group is that of stabilizer operators of the GKP code, eq. (2.27).

$$\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} (\hat{S}_q)^{n_1}(\hat{S}_p)^{n_2} \left|\psi^{\Delta}_{gkp}\right\rangle \left\langle\psi^{\Delta}_{gkp}\right| (\hat{S}^{\dagger}_q)^{n_1}(\hat{S}^{\dagger}_p)^{n_2}. \tag{2.34}$$

Then we expand the individual coherent states as in eq. (2.31):

$$\begin{aligned}\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} \int d^2\alpha d^2\beta \, \mathsf{exp}\left[-\frac{|\alpha|^2+|\beta|^2}{2\sigma^2_{gkp}}\right] \\ \times (\hat{S}_q)^{n_1}(\hat{S}_p)^{n_2}\hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\beta)(\hat{S}^{\dagger}_q)^{n_1}(\hat{S}^{\dagger}_p)^{n_2}.\end{aligned} \tag{2.35}$$

Because the stabilizer operators from eq. (2.27) are just specific displacement operators, we can write them as: $\hat{S}_q = \hat{D}(i\sqrt{2\pi}), \hat{S}_p = \hat{D}(\sqrt{2\pi})$. Now one can make use of an identity based on the Baker-Campbell-Hausdorff formula which states: $\hat{D}(\alpha)\hat{D}(\beta) = \hat{D}(\beta)\hat{D}(\alpha)e^{\alpha\beta^*-\alpha^*\beta}$. When switching the positions of the stabilizers and the integrated displacement operators, the following exponentials arise:

$$\begin{aligned}(\hat{S}_q)^{n_1}(\hat{S}_p)^{n_2}\hat{D}(\alpha) &= \hat{D}(in_1\sqrt{2\pi})\hat{D}(n_2\sqrt{2\pi})\hat{D}(\alpha) \\ &= \hat{D}(in_1\sqrt{2\pi})\hat{D}(\alpha)\hat{D}(n_2\sqrt{2\pi})e^{n_2\sqrt{2\pi}\alpha^*-n_2\sqrt{2\pi}\alpha} \\ &= \hat{D}(\alpha)\hat{D}(in_1\sqrt{2\pi})\hat{D}(n_2\sqrt{2\pi})e^{n_2\sqrt{2\pi}\alpha^*-n_2\sqrt{2\pi}\alpha+in_1\sqrt{2\pi}\alpha^*+in_1\sqrt{2\pi}\alpha} \\ &= \hat{D}(\alpha)\hat{D}(in_1\sqrt{2\pi})\hat{D}(n_2\sqrt{2\pi})e^{in_1\sqrt{2\pi}\,\mathsf{Re}(\alpha)-in_2\sqrt{2\pi}\,\mathsf{Im}(\alpha)}.\end{aligned} \tag{2.36}$$

An analogous reasoning can be made for $\hat{D}^{\dagger}(\beta)(\hat{S}^{\dagger}_q)^{n_1}(\hat{S}^{\dagger}_p)^{n_2}$, and the resulting total state is now:

$$\begin{aligned}\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} \int d^2\alpha d^2\beta \, \mathsf{exp}\left[-\frac{|\alpha|^2+|\beta|^2}{2\sigma^2_{gkp}}\right] \mathsf{e}^{i\sqrt{2\pi}(\mathsf{Re}(\alpha)-\mathsf{Re}(\beta))n_1-i\sqrt{2\pi}(\mathsf{Im}(\alpha)-\mathsf{Im}(\beta))n_2} \\ \times \hat{D}(\alpha)(\hat{S}_q)^{n_1}(\hat{S}_p)^{n_2} \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| (\hat{S}^{\dagger}_q)^{n_1}(\hat{S}^{\dagger}_p)^{n_2}\hat{D}^{\dagger}(\beta).\end{aligned} \tag{2.37}$$

Since the stabilizers, by definition, do not alter the ideal GKP states, they can be dropped:

$$\begin{aligned}\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} \int d^2\alpha d^2\beta \, \mathsf{exp}\left[-\frac{|\alpha|^2+|\beta|^2}{2\sigma^2_{gkp}}\right] \mathsf{e}^{i\sqrt{2\pi}(\mathsf{Re}(\alpha)-\mathsf{Re}(\beta))n_1-i\sqrt{2\pi}(\mathsf{Im}(\alpha)-\mathsf{Im}(\beta))n_2} \\ \times \hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\beta).\end{aligned} \tag{2.38}$$

Now the indices n_1, n_2 are only present in the scalar exponential function. This means one may invoke the summation formula of Poisson: $\sum_{n\in\mathbb{Z}} e^{ian} = 2\pi \sum_{k\in\mathbb{Z}} \delta(a-2\pi k)$, to obtain:

$$\begin{aligned}\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} \int d^2\alpha \, \text{exp}\left[-\frac{|\alpha|^2+|\beta|^2}{2\sigma^2_{gkp}}\right] \delta(\text{Re}(\alpha) - \text{Re}(\beta) - \sqrt{2\pi}k_1) \\ \times \delta(\text{Im}(\alpha) - \text{Im}(\beta) - \sqrt{2\pi}k_2) \hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\beta).\end{aligned} \tag{2.39}$$

The Dirac functions make the integrals equal to the value of the integrands at their peaks. This is equivalent to substituting $\text{Re}(\beta) = \text{Re}(\alpha) - \sqrt{2\pi}k_1$ and $\text{Im}(\beta) = \text{Im}(\alpha) - \sqrt{2\pi}k_2$ to give:

$$\begin{aligned}\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} \int d^2\alpha \, \text{exp}\left[-\frac{|\alpha|^2+|\alpha-\sqrt{2\pi}(k_1+ik_2)|^2}{2\sigma^2_{gkp}}\right] \\ \times \hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\alpha - \sqrt{2\pi}(k_1+ik_2)).\end{aligned} \tag{2.40}$$

One can split the exponential up:

$$\begin{aligned}\hat{\psi}^{\Delta}_{gkp} \propto \sum_{n_1,n_2\in\mathbb{Z}} \text{exp}\left[-\frac{\pi|k_1+ik_2|^2}{\sigma^2_{gkp}}\right] \int d^2\alpha \, \text{exp}\left[-\frac{|\alpha-\sqrt{\pi/2}(k_1+ik_2)|^2}{\sigma^2_{gkp}}\right] \\ \times \hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\alpha - \sqrt{2\pi}(k_1+ik_2)).\end{aligned} \tag{2.41}$$

From the exponential decay in the prefactor it is clear that all terms where $(k_1, k_2) \neq (0,0)$ are negligible for realistically small values of σ_{gkp}, and so we keep

$$\hat{\psi}^{\Delta}_{gkp} \propto \int d^2\alpha \, \text{exp}\left[-\frac{|\alpha|^2}{\sigma^2_{gkp}}\right] \hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\alpha), \tag{2.42}$$

since it is only a proportionality, we are free to add a factor:

$$\hat{\psi}^{\Delta}_{gkp} \propto \int \frac{d^2\alpha}{\pi\sigma^2_{gkp}} \, \text{exp}\left[-\frac{|\alpha|^2}{\sigma^2_{gkp}}\right] \hat{D}(\alpha) \left|\psi_{gkp}\right\rangle \left\langle\psi_{gkp}\right| \hat{D}^{\dagger}(\alpha) = \mathcal{N}[\sigma_{gkp}](\hat{\rho}), \tag{2.43}$$

which is exactly what we have in eq. (2.33).

2.3.5 Stabilizing error variance with a continuous supply of GKP states

This section explains the workings of GKP error correction in more rigorous detail. First, we recall some basic facts to support this derivation. The $SUM_{1\to 2}$ gate is denoted as:

$$SUM_{1\to 2} = e^{-i\hat{q}_1\hat{p}_2}. \tag{2.44}$$

An exponential operator linear in one quadrature effects a shift in eigenstates of the opposite quadrature as:

$$e^{-im\hat{p}}\left|r\right\rangle_q = \left|r+m\right\rangle_q, \tag{2.45}$$

while a global phase in eigenstates of its own:

$$e^{-im\hat{p}}\left|r\right\rangle_p = e^{-imr}\left|r\right\rangle_p, \tag{2.46}$$

Since we can expand ideal GKP states in terms of quadrature eigenstates:

$$\begin{aligned} e^{-im\hat{p}}\left|0\right\rangle_{gkp} &= e^{-im\hat{p}}\sum_{n\in\mathbb{Z}}\left|2n\sqrt{\pi}\right\rangle_q \\ &= \sum_{n\in\mathbb{Z}} e^{-im\hat{p}}\left|2n\sqrt{\pi}\right\rangle_q \\ &= \sum_{n\in\mathbb{Z}}\left|2n\sqrt{\pi}+m\right\rangle_q. \end{aligned} \tag{2.47}$$

We are now equipped to consider a case where the GKP gadget is applied to a noisy input state and with a noisy ancilla, as pictured in fig. 2.14.
The noise and the $SUM_{1\to 2}$ operations that act on the ideal GKP states are as follows:

$$e^{-i\hat{q}_1\hat{p}_2}e^{-iu_1\hat{p}_1}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2}. \tag{2.48}$$

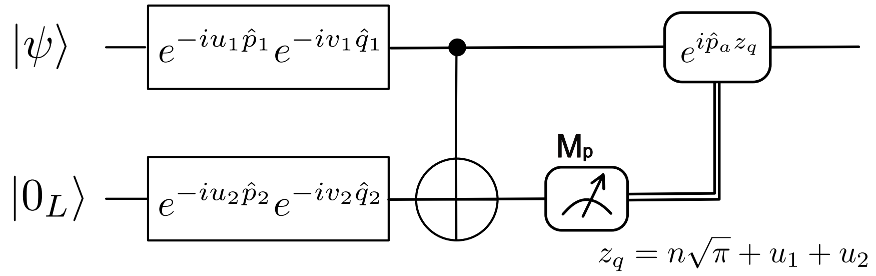

Figure 2.14: Circuit diagram showing the single noise scenario, where the input states are perfect GKP states, and all operations after the noise introduction are free of noise.

The $SUM_{1\to 2}$ operator can be shifted to the right, taking into account the commutation relations. The first shift goes as follows:

$$\begin{aligned} e^{-i\hat{q}_1\hat{p}_2}e^{-iu_1\hat{p}_1} &\overset{\text{BCH}}{=} e^{-i(u_1\hat{p}_1+\hat{q}_1\hat{p}_2)+\frac{1}{2}[-i\hat{q}_1\hat{p}_2,-iu_1\hat{p}_1]+\dots} \\ &= e^{-i(u_1\hat{p}_1+\hat{q}_1\hat{p}_2)-i\frac{u_1}{2}\hat{p}_2} \\ &= e^{-i(u_1\hat{p}_1+\hat{q}_1\hat{p}_2)+i\frac{u_1}{2}\hat{p}_2-iu_1\hat{p}_2} \\ &= e^{-iu_1\hat{p}_1}e^{-i\hat{q}_1\hat{p}_2}e^{-iu_1\hat{p}_2}, \end{aligned} \tag{2.49}$$

where in the first line, the Baker-Campbell-Hausdorff formula and eq. (2.20) are invoked. The next two shifts are easy since $[e^{-i\hat{q}_1\hat{p}_2}, e^{-iu_2\hat{p}_2}] = [e^{-i\hat{q}_1\hat{p}_2}, e^{-iv_1\hat{q}_1}] = 0$. Then the last shift goes as:

$$\begin{aligned} e^{-i\hat{q}_1\hat{p}_2}e^{-iv_2\hat{q}_2} &\overset{\text{BCH}}{=} e^{-i(v_2\hat{q}_2+\hat{q}_1\hat{p}_2)+\frac{1}{2}[-i\hat{q}_1\hat{p}_2,-iv_2\hat{q}_2]+\dots} \\ &= e^{-i(v_2\hat{q}_2+\hat{q}_1\hat{p}_2)+i\frac{v_2}{2}\hat{q}_1} \\ &= e^{-i(v_2\hat{q}_2+\hat{q}_1\hat{p}_2)-i\frac{v_2}{2}\hat{q}_1+iv_2\hat{q}_1} \\ &= e^{-iv_2\hat{q}_2}e^{-i\hat{q}_1\hat{p}_2}e^{iv_2\hat{q}_1}. \end{aligned} \tag{2.50}$$

The result is:

$$\begin{aligned} SUM_{1\to 2}e^{-iu_1\hat{p}_1}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2} &= e^{-iu_1\hat{p}_1}e^{-iu_1\hat{p}_2}e^{-iv_1\hat{q}_1}e^{-iu_2\hat{p}_2}e^{-iv_2\hat{q}_2}e^{-iv_2\hat{q}_1}SUM_{1\to 2} \\ &= e^{-iu_1\hat{p}_1}e^{-i(v_1-v_2)\hat{q}_1}e^{-i(u_2+u_1)\hat{p}_2}e^{-iv_2\hat{q}_2}SUM_{1\to 2}. \end{aligned} \tag{2.51}$$

From this, recalling eq. (2.45), we conclude that the effect of a noiseless GKP gadget can be summarized in a vector as follows:

$$\begin{pmatrix} \hat{q}_1 \\ \hat{p}_1 \\ \hat{q}_2 \\ \hat{p}_2 \end{pmatrix} \to \begin{pmatrix} \hat{q}_1 \\ \hat{p}_1 - \hat{p}_2 \\ \hat{q}_2 + \hat{q}_1 \\ \hat{p}_2 \end{pmatrix}. \tag{2.52}$$

Now the actual correction step consist of performing $e^{-iz_q\hat{p}_1}$, where $z_q = n\sqrt{\pi} + u_1 + u_2$ is the measurement result of a homodyne measurement of $\hat{q}$ of the ancilla qubit making the end state:

$$\begin{aligned} e^{-iz_q\hat{p}_1}e^{-iu_1\hat{p}_1}e^{-i(v_1-v_2)\hat{q}_1}\left|\psi\right\rangle &= e^{-i(n\sqrt{\pi}+u_1+u_2)\hat{p}_1}e^{-iu_1\hat{p}_1}e^{-i(v_1-v_2)\hat{q}_1}\left|\psi\right\rangle \\ &= e^{-iu_2\hat{p}_1}e^{-i(v_1-v_2)\hat{q}_1}\left|\psi\right\rangle. \end{aligned} \tag{2.53}$$

The original noise u_1 on the $\hat{q}$ quadrature of the input state is replaced by u_2 from the ancillary qubit. With a second ancilla qubit and a $SUM^{\dagger}_{1\to 2}$ gate, the $\hat{p}$ quadrature can be corrected in the same way. This means that with access to a stream of fresh GKP ancillaries, we can control the variance of the input state to this of the ancillas, which is the GKP gadget's real power.

2.4 Concatenation of GKP and surface code

The main shortcoming of the GKP code is that once a quadrature has crossed the threshold of $\frac{\sqrt{\pi}}{2}$, it will be stabilized to the opposite logical state and can never be retrieved. We can say: ”The GKP code cannot correct over the boundaries of bins.” By concatenating with another code, an idea that first came up in ref. [20], we can supersede this capacity and correct over the boundaries of bins. The rotated surface code was proposed to serve this purpose [1]. In fig. 2.15 a schematic of the concatenated code is shown, along with a clarification in fig. 2.16. Note that the X-type syndrome measurements consist of alternating sum and inverse sum gates. This is an enhancement of the naive code, where all are $SUM_{j\to k}$ gates, because it avoids a specific noise amplifying error mode that is not uncommon, namely the propagation of an error on a syndrome ancillary to a neighboring syndrome ancillary (of the other quadrature). For details, see ref. [1], Fig. 5. This scheme relies on the fact that, for GKP qubits, the effect of $SUM_{j\to k}$ and $SUM^{\dagger}_{j\to k}$ is the same in the code space but not outside the code space.
Concatenation of the two codes improves the overall performance. The GKP error correction is applied to stabilize the physical qubits individually between each round of the surface code.

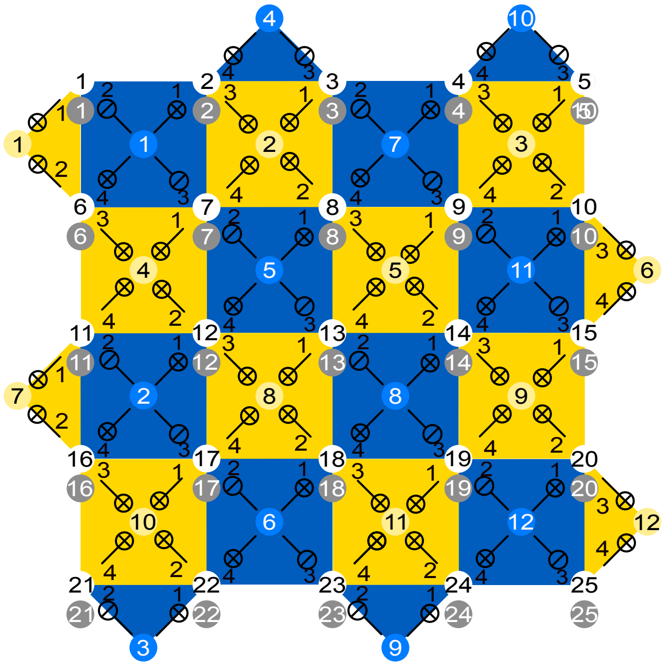

Figure 2.15: The Surface-GKP code for d=5. The data qubits are in white circles, grey circles hold the ancillaries for GKP stabilization. The surface code ancillaries are in blue and yellow circles for X-type and Z-type syndromes respectively.

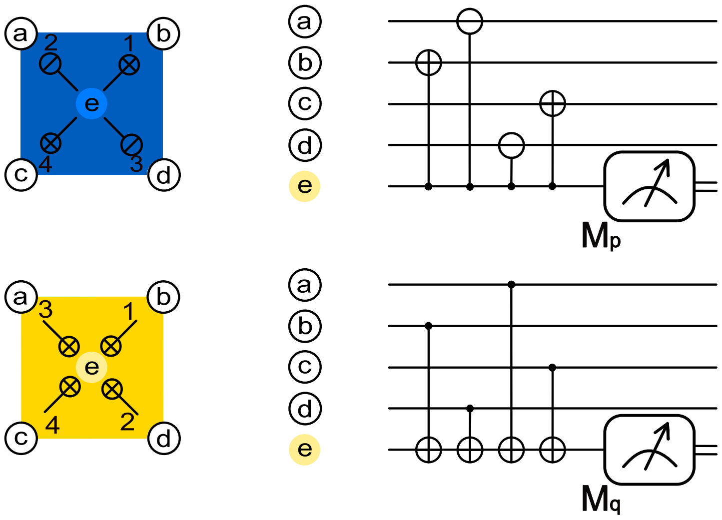

Figure 2.16: Circuit excerpts from fig. 2.5 with their translation in conventional circuit notation to show more explicitly how the notation works. The indices within the squares refer to the order in which the SUM and inverse SUM gates are performed.

However, there is some outstanding optimization. The two codes can synergize when one uses the ”analog information” from the syndrome measurements during the GKP stabilization. With this information, we can improve the estimate of the probability that the data qubit underwent an error. When decoding the surface code, this probability is directly integrated into the weights of the graph, as we saw in section 2.2.3. The increased accuracy of this estimate improves error-correcting capability of the concatenated code [1, 96, 97, 22, 28].

The fact that the standard assumption of stochastic i.i.d. noise on each physical qubit is not applicable for bosonic codes [2] makes the definition of distance harder. It is no longer just the weight of the minimal logical operator that is non-trivial. Ref. [2] proposes a new definition of ”Euclidean distance” based on the generalization of GKP as a stabilizer code.

We know the history of Hamiltonians a qubit underwent, resulting in an analytical estimate of the error probability. The expected error variances are calculated by carefully keeping track of the error propagation throughout the circuit. However, on top of that, we now have an indicative value as to its actual (modular) quadrature displacement value. The ancillary measurement from the GKP stabilization gives exactly that modular displacement of the corresponding data qubit, up to some noise. Incorporating this knowledge increases the likelihood that the right qubits are corrected and thus improves the overall performance of the code, as we will show in section 3.2. This incorporation is done by using the following formula for the probability that a data qubit underwent an error if its expected error variance is σ^2 and the corresponding GKP ancillary measured a modular quadrature displacement value of z:

$$p[\sigma](z) = \frac{\sum_{n\in\mathbb{Z}} \exp\left[-(z-(2n+1)\sqrt{\pi})^2/(2\sigma^2)\right]}{\sum_{n\in\mathbb{Z}} \exp\left[-(z-n\sqrt{\pi})^2/(2\sigma^2)\right]}. \tag{2.54}$$

This formula represents the normalized probability that the measured value z is part of the logical $|1\rangle$ distribution of Gaussian peak of variance σ^2 and has therefore undergone an error assuming that the initial state is logical $|0\rangle$. The denominator is the probability that the measured value z is part of the code space at all, belonging to either logical $|0\rangle$ or logical $|1\rangle$. Good intuition for this can result from inspecting fig. 2.17. These show clearly that if a quadrature measurement is close to the decision border of $\frac{\sqrt{\pi}}{2}$, that this physical qubit has very likely undergone an error. Within the MWPM decoding framework, this means we want to give the edge that corresponds to this physical qubit a very low weight, making it very likely that it will be picked up as part of an error chain. They also show that physical qubits, which we expect to show a large displacement due to their history, are less trusted. So even if their ancillary measurement shows no sign of an error, the corresponding weight will still be relatively low. Note that, in section 3.3, we will see that these probabilities are translated to weight with a logarithmic function. This means that the differentiating power in the curves in fig. 2.17 is softened.

Knowing the correct expected error variance of a certain data qubit is instrumental. It depends on the position and parity of the index, but also the round of syndrome measurement.

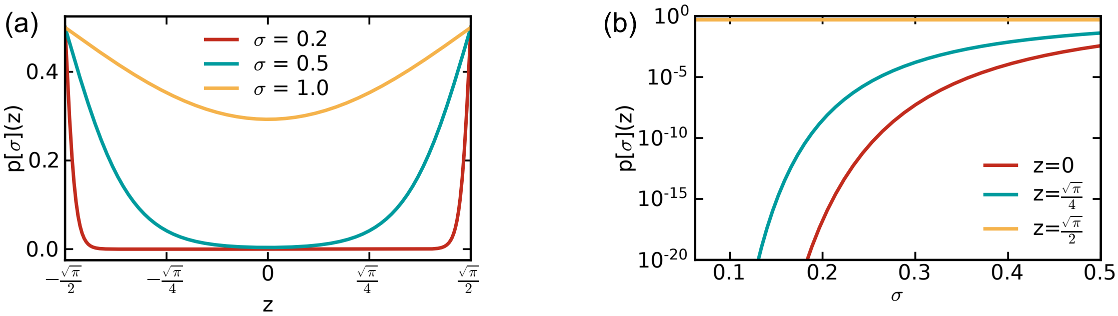

Figure 2.17: (a) Error probability as function of the measured quadrature for several values of the error variance. As expected, the error probability increases with variance σ^2 and the dependence on the measured value is very sensitive to σ^2. (b) The same error probability but as a function of the error variance for several values of measured quadrature.

The weights and error variances are stated in section 3.3 and are part of the new results in this thesis. The following section 2.4.1 will be devoted to detailing the noise model within which we will probe the concatenated code.

2.4.1 Noise Model

Generally, one assumes that small displacements are more likely to occur than large displacements. One probability distribution that has this feature is the Gaussian distribution. The use of Gaussian distributions to model the randomness of errors in the GKP code is a conventional practice [29, 22, 1, 98].

In addition to the noise coming from imperfect GKP resource states, due to finite σ_{gkp} in eq. (2.33), three other noise sources are considered. Fig. 2.18 shows the four noise sources included in the model. Both the GKP code and surface code can be described by the GKP gadget, as the stabilizer operations in the surface code are a generalization where more than one $SUM_{j\to k}$ is performed before applying any correctional shifts.

In steady state operation, the only continuous influx of errors due to finite GKP squeezing comes from the fresh ancillary qubits. Therefore, any time a new ancilla is generated, it is initialized with random displacement according to a Gaussian distribution $\mathcal{N}(0,\sigma_{gkp}^2)$. During execution of the $SUM_{j\to k}$ gates, the two involved qubits will stand to incur correlated noise, modeled by a bi-variate normal distribution with a prefactor σ_{sum}^2 in the covariance matrix, which is found in Appendix A of ref. [1]. Anytime a homodyne measurement is performed, an opportunity for noise arises. We model this again as a Gaussian displacement channel with a variance σ_{meas}^2 [1]. One thing that sets apart data qubits from ancillaries is that the former will sometimes be idling. Most notably, each quadrature will incur idling noise while the other is undergoing its GKP stabilization procedure, but also during the surface code procedure some data qubits will be idling as a boundary effect. During idling, bosonic states are vulnerable to photon loss. Ref. [1] shows that a conservative approximation for photon noise is a random displacement error channel, obtained by adding a heating noise term. The corresponding variance is coined σ_{photon}^2. The only components assumed to be noiseless are the displacement operators used to correct the measured shift because they do not have to be physically implemented since one can keep track of them in the Pauli frame [1, 99, 100, 70, 81].

A sensible grouping is in three classes, based on which types of qubits they affect, going by ref. [50]. ”Class-0” holds all noise sources that only directly affect the data qubits, $\sigma_0 = \sigma_{photon}$. "Class-1" holds those that affect only the ancillary qubits directly, $\sigma_1 = \sigma_{gkp} = \sigma_{meas}$. Here, it is important to recognize that σ_{meas} does not act on the data qubits during steady state operation but only when the logical state might eventually be read out. Finally, "class-2" contains those that affect both simultaneously through two-qubit gates, $\sigma_2 = \sigma_{sum}$.

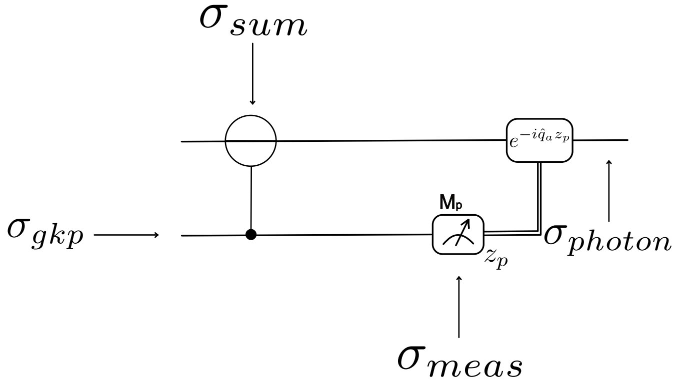

Figure 2.18: Schematic of where the four error sources come in. The ancilla will be initialized with a finite squeezing, resulting in a variance σ_{gkp}. The $SUM_{j\to k}$ gates will introduce correlated noise with a variance σ_{sum}. Homodyne measurements come with a noise variance of σ_{meas}. Finally, whenever a qubit is not undergoing an operation or measurement, it will suffer from idling noise, with a variance σ_{photon}.

3

Main Results

This chapter contains all the new results in this thesis. First, section 3.1 clarifies the strategy for simulation and extraction of results. Then, section 3.2 addresses a reproduction of results from the original work [1] to verify the framework.
After that, all results are new. Section 3.3 states the expanded expressions for the expected variance of physical qubits and weight for the decoding graph. Section 3.4 compares the surface-GKP code to some other concatenated codes involving the GKP code and several topological stabilizer codes. Sections 3.5 through 3.9 treat the probation of the rotated surface-GKP code in different scenarios. Except for section 3.7, which treats an interpretation of the newfound phenomenon that became apparent from this exploration, a proposed equivalent information flow model.

3.1 Simulation and result-extraction details

The simulations are performed in a Monte Carlo strategy within the analytical noise model, based on a twirling approximation. This method is standard for determining the threshold of concatenated codes consisting of the GKP code and a stabilizer code [23, 1, 101, 102, 22]. Only the noise model that one considers is often different. The error correction procedure is performed on a set of samples while drawing random displacements from the appropriate distributions. After classifying each one as corrected or not, as explained in section 2.2.3, we obtain an error rate as the fraction of uncorrected samples. Since every observation is a Bernoulli experiment, either a logical error or not, the error rate, which is the average of these, naturally follows the binomial distribution. Therefore, we may use eq. (3.1) to quantify the standard deviation of the error rate estimate:

$$\sqrt{E\left[(P_L - E[P_L])^2\right]} = \sqrt{\frac{P_L(1-P_L)}{N}}, \tag{3.1}$$

where P_L is the ratio of logical errors in a sample, and N is the sample size (the number of simulations run). The error bars are symmetric, and we take one standard deviation on each side, corresponding to a confidence interval of 68.2%.

The threshold is defined as the noise level under which the logical error rate starts decreasing with increasing distance of the surface code. Because of boundary effects, this threshold is not a constant value but instead depends on d itself. The most rigorous definition is when we require d$\rightarrow\infty$.

One technique [103], which copes with the dependence of the threshold on d, is to fit the error rates from all distances locally to one quadratic function $y = ax^2+bx+c$, where the physical error (σ) is re-scaled for each distance as $(\sigma - \sigma^*_\infty)d^{1/\eta}$. The five parameters, $(a, b, c, \eta, \sigma^*_\infty)$, would be determined with Jack-knife resampling [104] and one would thus find an estimate of the threshold at infinity. Since we can only simulate low dimensions in a reasonable time frame, this technique is not applicable. Therefore, this thesis quantifies the threshold as the crossing of the error rates for $d = 5$ and $d = 7$. Notice that the crossing results from linear interpolation in the logarithmic scale. The black lines on fig. 3.1 illustrate this process.

In section 3.9, we need to be able to quantify the uncertainty on the threshold estimate. Graphically, Fig. 3.1 shows how the green lines determine the upper quantile and the red ones the lower quantile.

Usually, the Monte Carlo simulation is stopped when 500 logical Z errors have been simulated. When the error rate is very low, this is often impossible in a realistic time frame for codes of distance larger than $d = 3$.

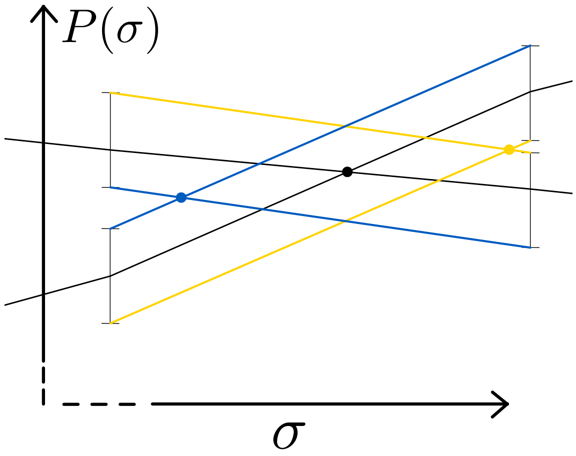

Figure 3.1: Illustration of the calculation of the uncertainty on the threshold estimate using the error bars of the error rates. The black lines are the interpolated error rate estimates, whose crossing is the estimate of the error threshold. The yellow lines determine the upper quantile of the threshold estimate, and the blue ones the lower quantile. Notice that the y axis shows a logarithmic scale.

3.2 Reproduction of previous results

The work in this thesis is built on the framework provided by ref. [1]. The code for their project is not public, so a considerable amount of effort and time was spent reconstructing the code framework based on the pseudo-code in the appendices of the original work. To verify the validity of the code, we reproduce their results in fig. 3.2.

We observe a slight disagreement with the original work with regard to the thresholds. Their threshold is calculated as the crossing of $d = 7$ and $d = 9$, while ours is that of $d = 5$ and $d = 7$. Additional potential reasons for the disagreement are the numerical conditioning of the problem of obtaining the threshold since the error rate curves are quite parallel. On top of that, the accuracy while calculating the weights is limited.

The original work explores only a small part of the parameter space. More specifically, three one-dimensional slices through the four-dimensional parameter space. The framework allows for simulation of both the cases with and without incorporation of the analog information from ancillary measurement into the calculation of weights for the MWPM. Respectively eq. (2.54) and eq. (2.32) determine the weights in the decoding graph. Section 3.1 detailed the determination of the threshold. In fig. 3.2(a), we first study the case I. This is a simulation where we only vary σ_{gkp} and keep the generalized circuit noise at $\sigma_m = \sigma_p = \sigma_c = \sigma = 0$. The threshold (with analog information) is found at $\sigma^* = 0.183$. In fig. 3.2(b), the general circuit noise(σ) is varied, while σ_{gkp} is kept at 0. The threshold is now found at $\sigma^* = 0.085$. In fig. 3.2(c), both the general circuit noise (σ) and σ_{gkp} are varied simultaneously ($\sigma = \sigma_{gkp}$). The threshold is now found at $\sigma^* = 0.078$.

A curiosity, which will be addressed further in section 3.7, is that the threshold variances behave like the resistances in a parallel circuit.

$$\frac{1}{0.183}^{2} + \frac{1}{0.085}^{2} = \frac{1}{0.078}^{2} \tag{3.2}$$

Note that this also holds for the results in the original work:

$$\frac{1}{0.194}^{2} + \frac{1}{0.09}^{2} = \frac{1}{0.083}^{2} \tag{3.3}$$

A common feature in all results is that the error rates level off to $P = 0.25$ for large noise. It may be surprising to a classical thinker who expects $P = 0.5$ here. However, in contrast to a classical bit which can only undergo one kind of error (the bitflip), a qubit can undergo three different errors. In a purely random event, one of the four operations ($\hat{I}, \hat{X}, \hat{Y}, \hat{Z}$) happen with equal probability, $P = 0.25$ for each of them. Note that the error rates for X and Z are the same, and we will, from here on, refer to either as just (logical) error rate. In the limit of small error rates, we have $P_Y = P_Z * P_X = P^2$.

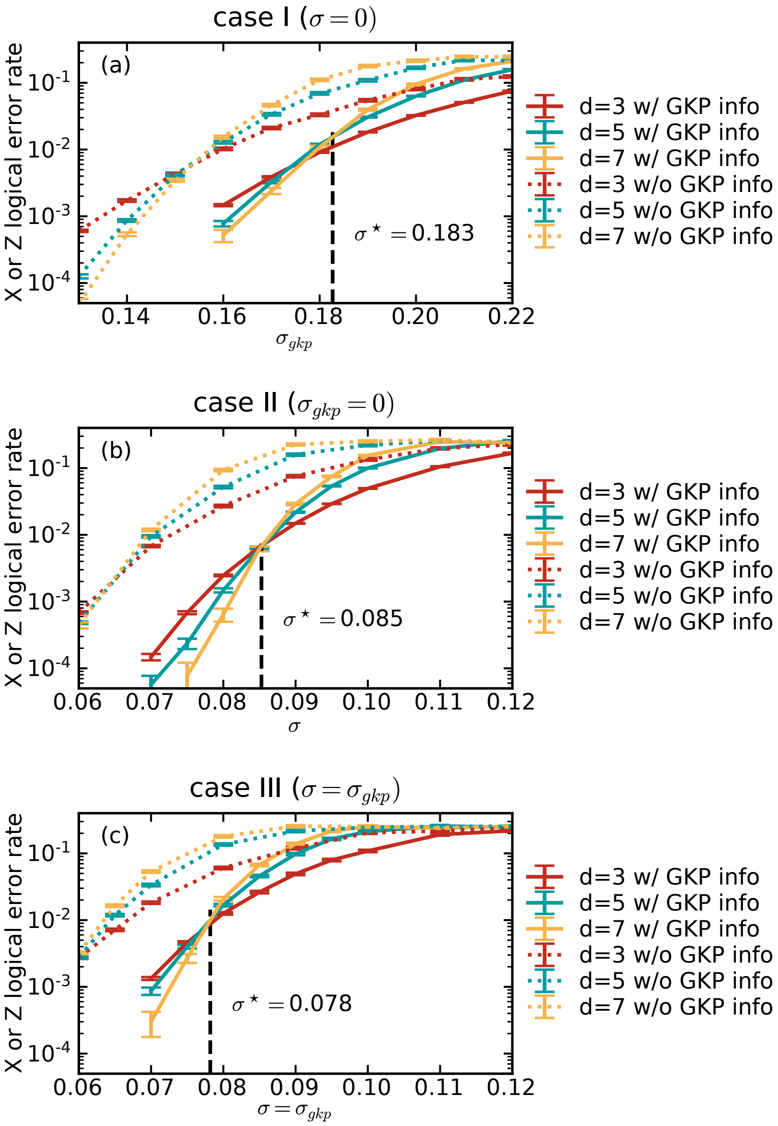

Figure 3.2: Simulation results of the logical error rate of the surface-GKP code for several distances. (a) Case I: σ_{gkp} is varied while the generalized circuit noise is disabled ($\sigma = 0$), (b) Case II: the generalized circuit noise, σ, is varied while the GKP state squeezing is infinite ($\sigma_{gkp} = 0$) and (c) Case III: both $\sigma = \sigma_{gkp}$ are varied simultaneously. The solid lines represent logical error rates when incorporating information from the GKP stabilizer measurements to re-normalize edge weights in the matching graphs. The dashed lines represent the same situations but without information from GKP stabilizer measurements.

The other extreme of the parameter space is also interesting to study. However, because of the inefficiency of the framework, simulations of very low error rates are not tractable. Therefore, studying how exactly the errors behave in that limit is unfeasible, within our current simulation framework. We observe something similar to an exponential drop, but there is no known function to which these error rates fit.

Finally, for all cases, the inclusion of the analog information is significantly boosting the threshold, and the error rates are decreased by at least an order of magnitude at the threshold.

3.3 Weights

The expressions reported in this section were derived by my supervisor, Timo Hillmann. They result from a careful study of how the noise propagates throughout the code. In the following, k is the qubit index as in fig. 2.15 and d is the distance of the $d \times d$ rotated surface code.

The probability that a GKP ancilla has undergone a logical GKP error determines the weights for the corresponding horizontal edges of the graph in the decoder. The two quadratures are inevitably corrected consecutively. The order of the corrected quadratures alternates between qubit indices. Therefore, the quadratures will have a different variance, depending on whether it was the first or second to be stabilized on a particular qubit. We first consider the weights corresponding to the quadrature that was corrected earliest:

$$w_Z^H(k;d) \leftarrow \begin{cases} -\log_2\left(p\left[\sigma_{1,1}^Z\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Ak)}\right)\right)\right) & \text{round 1} \\ -\log_2\left(p\left[\sigma_Z^H(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Ak)}\right)\right)\right) & \text{round 2 to d} \\ -\log_2\left(p\left[\sigma_{1,d+1}^Z(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Ak)}\right)\right)\right) & \text{round d+1,} \end{cases} \tag{3.4}$$

with

$$\begin{aligned} \sigma_{1,1}^Z &= \sqrt{\sigma_{gkp}^2 + \frac{4}{3}\sigma_{sum}^2 + \sigma_{meas}^2 + \sigma_{photon}^2}, \\ \sigma_{1,d+1}^Z(k;d) &= \sqrt{\sigma_Z^H(k;d) - \sigma_{gkp}^2 - \frac{4}{3}\sigma_{sum}^2 - \sigma_{meas}^2 - \sigma_{photon}^2}, \end{aligned} \tag{3.5}$$

for odd k, and

$$w_X^H(k;d) \leftarrow \begin{cases} -\log_2\left(p\left[\sigma_{1,1}^X\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)\right)\right) & \text{round 1} \\ -\log_2\left(p\left[\sigma_X^H(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)\right)\right) & \text{round 2 to d} \\ -\log_2\left(p\left[\sigma_{1,d+1}^X(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)\right)\right) & \text{round d+1,} \end{cases} \tag{3.6}$$

with

$$\begin{aligned}\sigma^X_{1,1} &= \sqrt{\sigma^2_{gkp} + \frac{4}{3}\sigma^2_{sum} + \sigma^2_{meas} + \sigma^2_{photon}},\\ \sigma^X_{1,d+1}(k;d) &= \sqrt{\sigma^H_X(k;d) - \sigma^2_{gkp} - \frac{4}{3}\sigma^2_{sum} - \sigma^2_{meas} - \sigma^2_{photon}},\end{aligned} \tag{3.7}$$

for even k. Here, $R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)$ is the measurement value of the $\hat{p}$ quadrature GKP ancillary qubit corresponding to data qubit k. $R_s(z) = z - s\lfloor \frac{z}{s} + \frac{1}{2} \rfloor$ is the modular function that simulates the modular measurement. The symbols $\xi_p^{(Ak)}$ and $\xi_q^{(Ak)}$ denote the bare $\hat{p}$ and $\hat{q}$ quadrature displacements respectively on the GKP ancillary qubit corresponding to data qubit k.
In this, $\sigma^H_X(k;d)$ and $\sigma^H_Z(k;d)$ are defined as:

$$\sigma^H_Z(k;d) = \begin{cases} \begin{cases} \sqrt{4\sigma^2_{gkp} + \frac{22}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 8\sigma^2_{photon}} & \frac{k-1}{d} \in 2\mathbb{Z} \\ \sqrt{4\sigma^2_{gkp} + \frac{31}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 7\sigma^2_{photon}} & \frac{k-1}{d} \in 2\mathbb{Z}+1 \end{cases} & k \in d\mathbb{Z}+1 \\ \begin{cases} \sqrt{4\sigma^2_{gkp} + \frac{19}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 10\sigma^2_{photon}} & \frac{k}{d} \in 2\mathbb{Z}+1 \\ \sqrt{4\sigma^2_{gkp} + \frac{22}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 7\sigma^2_{photon}} & \frac{k}{d} \in 2\mathbb{Z} \end{cases} & k \in d\mathbb{Z} \\ \sqrt{5\sigma^2_{gkp} + \frac{35}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 6\sigma^2_{photon}} & \text{otherwise,} \end{cases} \tag{3.8}$$

$$\sigma^H_X(k;d) = \begin{cases} \begin{cases} \sqrt{4\sigma^2_{gkp} + \frac{22}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 7\sigma^2_{photon}} & k \in 2\mathbb{Z}+1 \\ \sqrt{4\sigma^2_{gkp} + \frac{19}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 10\sigma^2_{photon}} & k \in 2\mathbb{Z} \end{cases} & k \in 1,...,d \\ \begin{cases} \sqrt{4\sigma^2_{gkp} + \frac{31}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 7\sigma^2_{photon}} & k \in 2\mathbb{Z}+1 \\ \sqrt{4\sigma^2_{gkp} + \frac{22}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 8\sigma^2_{photon}} & k \in 2\mathbb{Z} \end{cases} & k \in d^2-d+1,...,d^2 \\ \sqrt{5\sigma^2_{gkp} + \frac{35}{3}\sigma^2_{sum} + 2\sigma^2_{meas} + 6\sigma^2_{photon}} & \text{otherwise.} \end{cases} \tag{3.9}$$

The weights that can be assigned after step 2 are:

$$w^H_Z(k;d) \leftarrow \begin{cases} -\log_2\left(p\left[\sigma^Z_{2,1}\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Ak)}\right)\right)\right) & \text{round 1} \\ -\log_2\left(p\left[\sigma^H_Z(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Ak)}\right)\right)\right) & \text{round 2 to d} \\ -\log_2\left(p\left[\sigma^Z_{2,d+1}(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Ak)}\right)\right)\right) & \text{round d+1} \end{cases} \tag{3.10}$$

with

$$\begin{aligned}\sigma^Z_{2,1} &= \sqrt{2\sigma^2_{gkp} + \frac{8}{3}\sigma^2_{sum} + \sigma^2_{meas} + 3\sigma^2_{photon}},\\ \sigma^Z_{2,d+1}(k;d) &= \sqrt{\sigma^H_Z(k;d) - 2\sigma^2_{gkp} - \frac{8}{3}\sigma^2_{sum} - \sigma^2_{meas} - 3\sigma^2_{photon}},\end{aligned} \tag{3.11}$$

for even k, and

$$w_X^H(k;d) \leftarrow \begin{cases} -\log_2\left(p\left[\sigma_{2,1}^X\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)\right)\right) & \text{round 1} \\ -\log_2\left(p\left[\sigma_X^H(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)\right)\right) & \text{round 2 to d} \\ -\log_2\left(p\left[\sigma_{2,d+1}^X(k;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Ak)}\right)\right)\right) & \text{round d+1,} \end{cases} \tag{3.12}$$

with

$$\begin{aligned} \sigma_{2,1}^X &= \sqrt{\sigma_{gkp}^2 + \frac{8}{3}\sigma_{sum}^2 + \sigma_{meas}^2 + \sigma_{photon}^2}, \\ \sigma_{2,d+1}^X(k;d) &= \sqrt{\sigma_X^H(k;d) - 2\sigma_{gkp}^2 - \frac{8}{3}\sigma_{sum}^2 - \sigma_{meas}^2 - 3\sigma_{photon}^2}, \end{aligned} \tag{3.13}$$

for odd k.

Note that the weights are not space-time correlated, so from the second round on, the expressions do not change. Our scheme does not need space-time correlated edges, unlike when two-qubit gates are individually error corrected as in ref. [97]. GKP stabilization between rounds of the syndrome measurements makes sure that the variance of the qubits is the same after every round, as was explained in section 2.3.5. For the vertical edges, we consider the error probability of a surface syndrome:

$$w_Z^H(k;d) \leftarrow -\log_2\left(p\left[\sigma_Z^V(l;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_q^{(Zl)}\right)\right)\right), \tag{3.14}$$

$$w_X^H(k;d) \leftarrow -\log_2\left(p\left[\sigma_X^V(l;d)\right]\left(R_{\sqrt{\pi}}\left(\xi_p^{(Xl)}\right)\right)\right), \tag{3.15}$$

for $l \in \{1, ..., \frac{d^2-1}{2}\}$, the index of the surface syndrome qubits as in fig. 2.15.
Here, $R_{\sqrt{\pi}}\left(\xi_p^{(Xl)}\right)$ and $R_{\sqrt{\pi}}\left(\xi_q^{(Zl)}\right)$ are the modular measurement values of $\hat{p}$ and $\hat{q}$ quadratures of the stabilizers operators at index l.
The error probabilities $\sigma_X^V(l;d)$ and $\sigma_Z^V(l;d)$, with l is the index of the surface ancilla, are:

$$\sigma_Z^V(l;d) = \begin{cases} \sqrt{4\sigma_{gkp}^2 + \frac{20}{3}\sigma_{sum}^2 + 3\sigma_{meas}^2 + 9\sigma_{photon}^2} & l \in 2d''\mathbb{Z} + 1 \\ \sqrt{7\sigma_{gkp}^2 + \frac{50}{3}\sigma_{sum}^2 + 5\sigma_{meas}^2 + 14\sigma_{photon}^2} & l \in 2d''\mathbb{Z} + d'' + 1 \\ \sqrt{4\sigma_{gkp}^2 + \frac{37}{3}\sigma_{sum}^2 + 3\sigma_{meas}^2 + 9\sigma_{photon}^2} & l \in 2d''\mathbb{Z} \\ \sqrt{7\sigma_{gkp}^2 + \frac{65}{3}\sigma_{sum}^2 + 5\sigma_{meas}^2 + 12\sigma_{photon}^2} & \text{otherwise,} \end{cases} \tag{3.16}$$

$$\sigma_X^V(l;d) = \begin{cases} \sqrt{4\sigma_{gkp}^2 + \frac{20}{3}\sigma_{sum}^2 + 3\sigma_{meas}^2 + 9\sigma_{photon}^2} & l \in 2d''\mathbb{Z} + d'' \\ \sqrt{4\sigma_{gkp}^2 + \frac{37}{3}\sigma_{sum}^2 + 3\sigma_{meas}^2 + 9\sigma_{photon}^2} & l \in 2d''\mathbb{Z} + d'' + 1 \\ \sqrt{7\sigma_{gkp}^2 + \frac{50}{3}\sigma_{sum}^2 + 5\sigma_{meas}^2 + 14\sigma_{photon}^2} & l \in 2d''\mathbb{Z} \\ \sqrt{7\sigma_{gkp}^2 + \frac{65}{3}\sigma_{sum}^2 + 5\sigma_{meas}^2 + 12\sigma_{photon}^2} & \text{otherwise,} \end{cases} \tag{3.17}$$

where $d'' = \frac{d+1}{2}$. Note that it should be clear that the vertical weights use the GKP information from the surface syndrome qubit that is ”under” this edge, as the measurement needs to precede post-processing.

3.4 Comparison of a number of topological codes

This section compares several topological codes in concatenation with the GKP-code based on the following figures of merit. The performance, quantified as the threshold when all noise sources are enabled, is perhaps the most important but also the hardest figure to objectively compare because it depends to sensitively on the noise model under consideration. The number of steps in an error-correction cycle determines how long noise will accumulate in the system. The number of ancillary measurements affects the opportunities for noise to enter. Finally the algorithm used to decode the topological code will play a role in the performance and the time it takes to decode. We consider the color code on a square-octagon lattice (8,8,4 color code) [21], the toric code [22], the Surface-17 code with a minimal generating set [2] and the rotated surface-4 code [23], next to the rotated surface code [1]. In table 3.1 a list of features is shown. Note that d is the distance of the code.

Note that the thresholds should not be directly compared because the noise model is always different. All of them consider the information from the GKP syndromes to optimize the weights. The rotated surface code seems to come out very weak in this comparison, but this is because the error model is most extensive.

The simulation of the toric code [22] considers the oscillators undergoing Gaussian shift errors and the same type of noise during measurement. Within our framework, the closest equivalents are σ_{gkp} and σ_{meas}, even though the former is not an exact representation because ref. [22] do not consider propagation of shift errors [105].

Table 3.1: Comparison of different topological codes combined with the GKP-code

Method	Noise model[a]	Performance	Steps per error correction cycle	Number of ancillary measurements	Decoder
Rotated surface code	σ_{photon}, σ_{meas}, σ_{sum}, σ_{gkp}	threshold at $\sigma = 0.083$ if all σ are equal	6	$3d^2 - 1$	MWPM
Toric code	σ_{gkp}, σ_{meas}	$\sigma = 0.235$ if all σ are equal	6	3d^2	Maximum likelihood
8,8,4 color code	σ_{gkp}, σ_{meas}	$\sigma = 0.231$ if all σ are equal	10	2d$^2 + d^2/2$	MWPM
Surface-17 code	n/a[b]	n/a[b]	6	$2d^2$	Maximum likelihood
Rotated Surface-4 code (measurement based)	σ_{gkp}, σ_{sum}	$\sigma = 0.164$ for $\sigma_{sum} = 2\sigma_{gkp}$	12	$9d^2 - 1$	MWPM

---

[a] The variances reported here are ”converted” to match the parameters considered in this work. More explanation on this conversion can be found in the main text.

[b] Simulations of the threshold have not yet been performed for this protocol.

In the simulation of the color code, the GKP-qubits receive σ_{gkp}, and they consider two separate measurement noises, which we can only represent as σ_{meas}. The work on the surface-4 code considers σ_{gkp} and a gate noise which is not exactly the same as σ_{sum} because it is a measurement-based protocol. The number of steps per error correction cycle is very influential because it determines how long the system is ”open” for noise build-up. In order to counteract this, the surface-4 code performs multiple GKP error correction rounds during one surface code cycle. The idea is to limit the accumulation of noise during the surface code cycle. This positively affects the threshold, especially in decoder-limited architectures. By decoder-limited, we mean that the classical matching algorithm takes longer than the cycle of all quantum processes, leading to an unprotected stalling of the circuit.

Conversely, one could minimize the number of ancillary measurements in one cycle. A specialized algorithm proposed in ref. [2] calculates the minimal amount of coherent ancillary modes needed to perform the stabilizer measurements. This minimization is possible because GKP and surface stabilizers are not orthogonal, so measuring all is redundant. Some measurements do not add new information that was not implicitly present in the others. The surface-17 code is the minimal version of the rotated surface code, with $2d^2$ ancillary measurements. [2].

3.5 Influence of the four error sources separately

Here we discuss the results of simulations where only one of the error sources is considered at a time to compare how pervasively they individually affect the threshold and error rate. In fig. 3.3 the four cases are shown. The error rate is relatively similar at the threshold for all cases, if a bit lower for σ_{sum}. However, the threshold itself is very sensitive to the kind of noise. The equations for the expected variances in section 3.3 predict this. We see that for σ_{gkp} the threshold is distance dependent, while for the other noise sources the threshold seems fixed. Presumably, this phenomenon emerges through the different effects the noise sources have on boundary effects. The boundary data qubits only come in contact with σ_{gkp} twice or thrice instead of four times, thus decreasing the error rate for low distances and thereby shifting their thresholds downward. Another way this can be seen is that for a small surface code, the ratio $\frac{d^2-1}{d^2}$ is smaller than for large codes. In other words, there are relatively less noisy ancilla qubits on the grid, making the code perform better. In contrast, σ_{meas} shows no boundary effects at all since it affects each qubit equally, thus holding no dimensional nature. The more subtle cases are those of σ_{photon} and σ_{sum} because we expect a boundary effect since the qubits on the edges idle more but undergo less gates. However, fig. 3.3 shows no discernible influence of the distance on the threshold in either case. A plausible explanation is that the processes during the GKP stabilization, which are uniform, overshadow the boundary effects.

The order in size of the thresholds for the four noise sources agrees with the coefficients in the expressions in eqs. (3.9) and (3.17). The larger the coefficient, the more pervasive the noise and the smaller the threshold. The threshold for σ_{sum} is lower than those for the others, indicating that these components are the most critical for successful completion of the error correcting procedure. Intuitively this makes sense, because if we are not measuring the stabilizer operators anymore, then the whole concept of the code breaks down. The thresholds should be interpreted relative to how challenging the corresponding noises are to suppress.

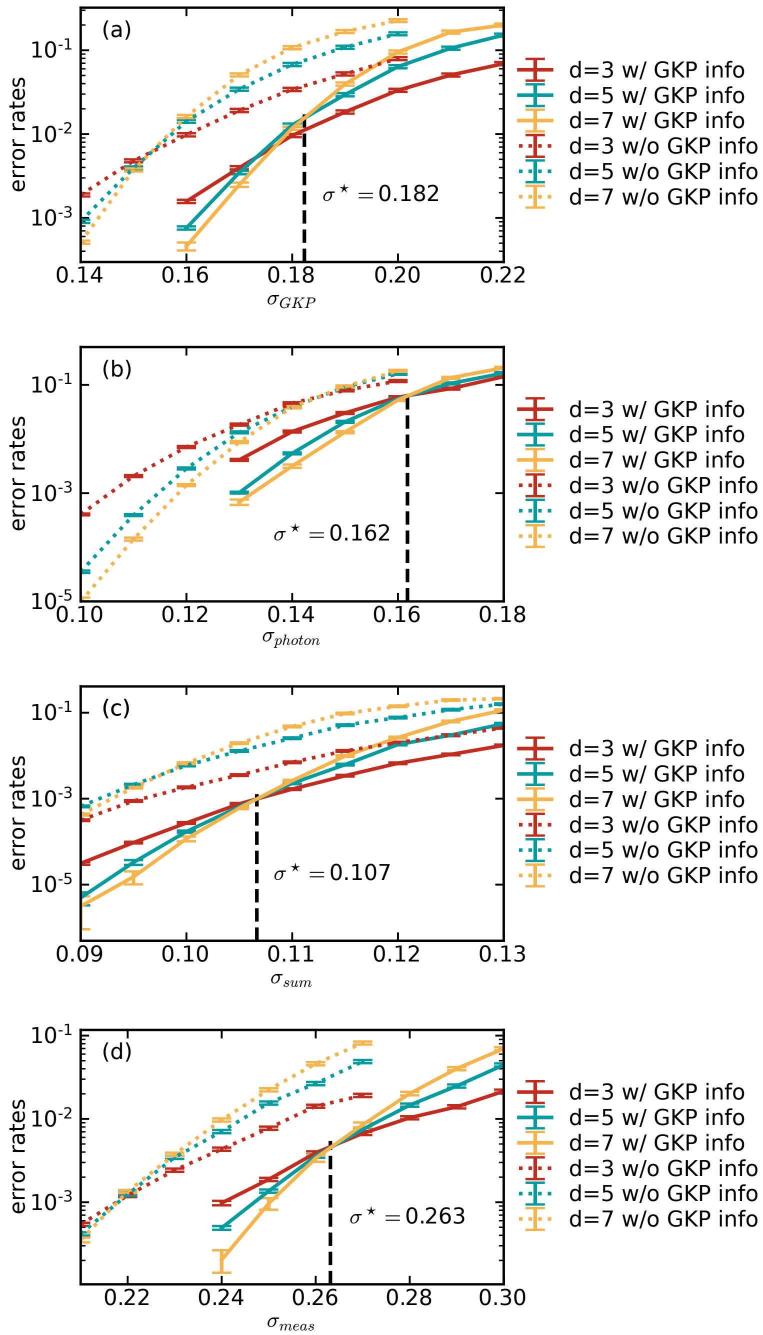

Figure 3.3: Logical error rate of the surface-GKP code for several distances. (a) Only the GKP squeezing is varied, while all other noise sources are disabled. (b) Only the idling noise is varied, while all other noise sources are disabled. (c) Only the SUM-gate noise is varied, while all other noise sources are disabled. (d) Only the measurement noise is varied, while all other noise sources are disabled.

3.6 Comparison with the discrete variable rotated surface code

Four different variances are considered in the noise model and grouped into three classes. Here we study the effect of these classes individually and combined on the error rate. Firstly we study the performance when only the separate error sources are present, in fig. 3.4.

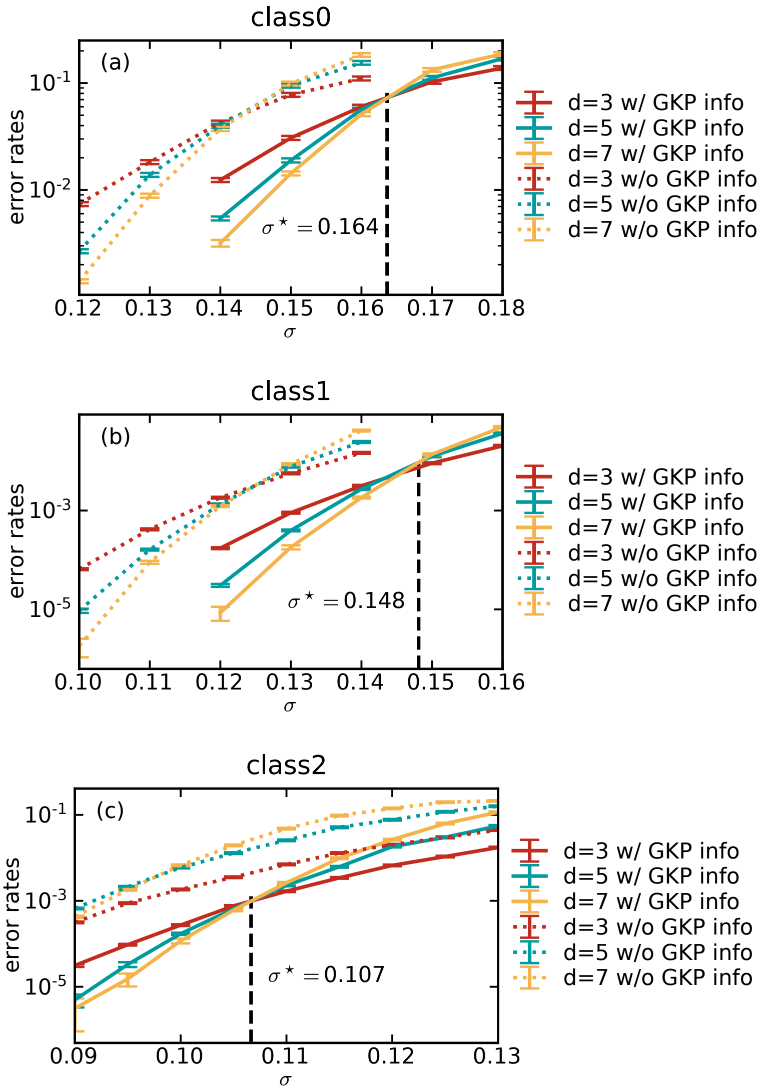

Figure 3.4: Logical error rate of the surface-GKP code for several distances. (a) class-0 noise is varied, while the other noise sources are disabled. (b) class-1 noise is varied, while the other noise sources are disabled. (c) class-2 noise is varied, while the other noise sources are disabled.

Since the class-0 and class-2 are the same as σ_{photon} and σ_{sum} which were discussed in section 3.2, we focus our attention to class-1, which is equivalent to $\sigma_{gkp} = \sigma_{meas}$. The threshold is lower than for the individual error sources, as expected. This mechanism is reminiscent of a parallel circuit, where two parallel, independent error modes lower the total ”resistance” for the ”error flow”. This analogy is discussed in more detail in section 3.7.

Studying the parameter space of the three noise sources is an elaborate task. Therefore it is necessary to approach it in a structured fashion. One could quantify the error correction threshold surface in the whole space. There are two arguments against this. Firstly, the quantification of the threshold is numerically poorly conditioned because the error rate curves are very close to parallel. Secondly, finding the threshold requires simulation of more than one dimension, while computational power is expensive since our framework is not efficient. Quantifying the crossing of the error rates with $P = 0.02$ for $d = 5$, instead of the threshold [50], solves both problems. This figure still gives a good sense of the performance of the code. However, this constant error surface is not under the threshold. In fig. 3.5, we observe an ellipsoid behavior of the contour lines, which a fit confirms. We fit the results to an ellipsoid function $\sum_{n=0}^{2} \frac{\sigma_n^2}{a_n} = 1$, with the result:

$$\left(\frac{\sigma_0}{0.200}\right)^2 + \left(\frac{\sigma_1}{0.127}\right)^2 + \left(\frac{\sigma_2}{0.125}\right)^2 = 1 \tag{3.18}$$

All error sources behave as parallel independent channels for ”error-flow” and are therefore interchangeable, albeit with an appropriate coefficient. In section 3.7, this argument is expanded. This behavior allows to develop predictive power of the performance of the code, which is desirable to have.

For the case without inclusion of GKP information, the fit comes to

$$\left(\frac{\sigma_0}{0.180}\right)^2 + \left(\frac{\sigma_1}{0.114}\right)^2 + \left(\frac{\sigma_2}{0.111}\right)^2 = 1 \tag{3.19}$$

As expected, the values of all vertices are smaller, as we see in fig 3.6. This confirms that the incorporation of analog information is always beneficial. When one would make this plot for a larger error rate, the ellipsoid volume would grow.

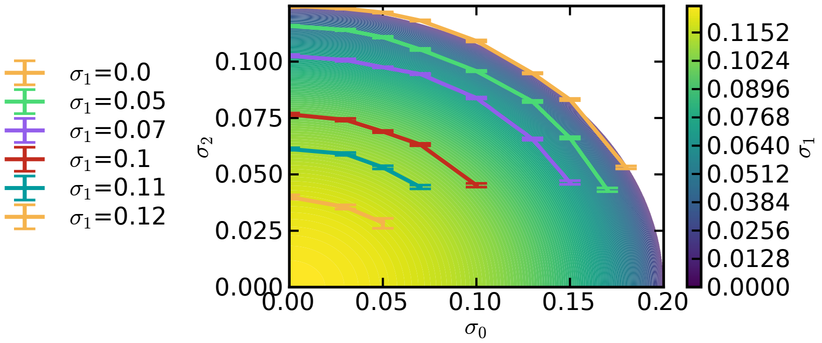

Figure 3.5: Numerical results variance combinations that correspond to an error rate of P=0.02 for $d = 5$, with the inclusion of analog information. The contour plot is an ellipsoid fit to this data. In the white area, the logical error rate is always above P=0.02. The numerical results are obtained by interpolating the logical error rate in σ_2 for each combination of (σ_1,σ_0) and finding the crossing with $P = 0.02$.

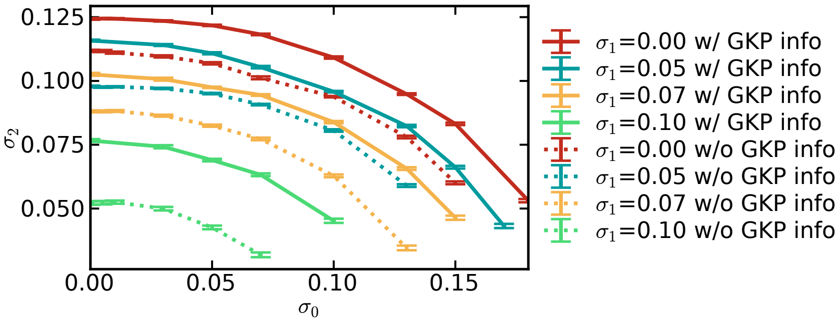

Figure 3.6: Numerical results variance combinations that correspond to an error rate of P=0.02 for $d = 5$. Dashed and full lines correspond to the case without and with the inclusion of analog information, respectively.

In Fig. 3.7 the results of a DV simulation in ref. [50] are shown. There is no obvious meaningful quantitative relation between the Gaussian standard deviation of the noise classes and a binary error rate used in DV systems. Therefore, we stick with reporting the standard deviation of the Gaussian noise instead. Nevertheless, an interesting feature we like to note, presumably caused by this ”conversion” of physical error rates, is that the constant error surface in the DV case is concave while it is convex in the CV case.

Despite having no way to compare the absolute numbers, we can compare the relative behavior of the error sources. Both our results and those from [50] show that the class-2 errors are most detrimental.

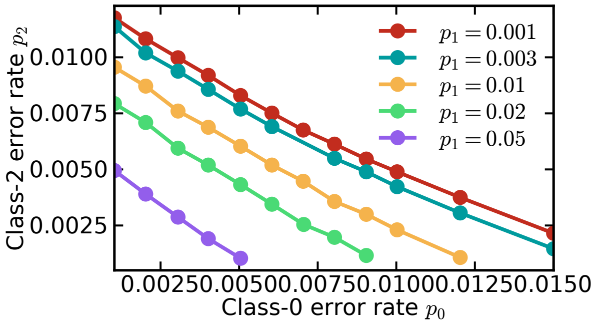

Figure 3.7: [data recovered from fig.8 in ref. [50]] A contour plot of the combination of error rates from the 3 error classes that correspond to a logical error rate of P=0.02 in a DV surface code.

This feature is expected since the sum gates are at the core of the error correction mechanism. Naturally, this holds for both DV and CV implementations. The DV case relatively allows more class-1 errors. The most likely explanation is the introduction of bosonic qubits, which brings in the σ_{gkp}, which is absent in DV. Generally, the noise model is more elaborate in the CV simulation. The GKP stabilization in the CV implementation makes the total code relatively more resistant to the class-0 and class-2 errors. However, since there is no good way to compare the variances with the discrete error rates straightforwardly, we cannot confirm this from the results.

3.7 Error variances as resistances to information flow

Because the errors are Gaussian and uncorrelated[a], we have to add the variances to obtain the total variance, as in section 3.3. This means one can understand an error mode, or a variance thereof, as a resistor to information flow. As we found in section 3.6, this model fits the results. For a certain error rate, the error sources are interchangeable with certain factors, according to $\sum_n \frac{\sigma_n^2}{a_n} = 1$. In the case of the reproduction data from section 3.2, the error thresholds correspond to very comparable error rates, so we expect a reasonable agreement with this behavior. We only need 2 data points to fit the elliptic behavior, allowing one observation to check the hypothesis. We fit cases I and II into: $\left(\frac{\sigma_{gkp}}{0.183}\right)^2 + \left(\frac{\sigma}{0.085}\right)^2 = 1$. Case 3

---

[a] The sum noise creates a correlation between the two involved qubits, but here we are discussing a correlation between separate error sources.

corresponds to the case $\sigma = \sigma_{gkp}$. When substituting this into the elliptic function, one obtains that $\sigma = 0.078$, which is the observed value. Note that one could ask how the coefficients a_n depend on the error rate P. Since $a_k = \sigma_k$ when all the others $\sigma_{n \neq k} = 0$, which is the case at the vertices of the ellipsoid, they will evolve with the logical error rate in exactly the same way that we saw in fig. 3.4. Fig. 3.8 shows a verification of this, where we recognize the transpose of the usual graph of error rate as a function of the standard deviation of the noise.

We may compare this to an electrical circuit in fig. 3.9 where the current is a flow of information. The resistors are proportional to the variance of the Gaussian error $R \propto \sigma^2$, and the proportionality constant depends on the kind of error source. This coefficient can be modeled as a sub-circuit for each error mode, as they influence the procedure at different points and with different pervasiveness. Fig. 3.10 illustrates this in the dual picture, meaning we consider the error modes as conductances for error flow, connected in parallel.

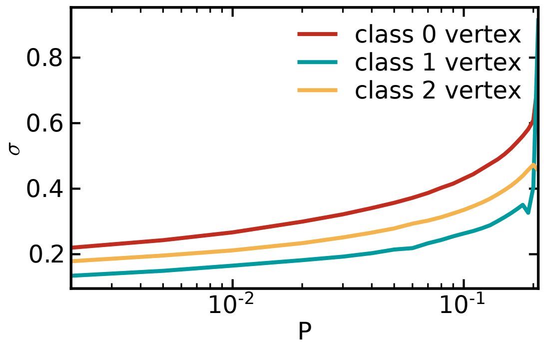

Figure 3.8: Values of the vertices in the ellipsoid fit for a single GKP mode as a function of the logical error rate. Note that the curves expectedly diverge at $P = 0.25$ towards the right side of the plot. Statistical fluctuations prevent a closer approach to the site of divergence. This plot qualitatively behaves as the transpose of fig. 3.4.

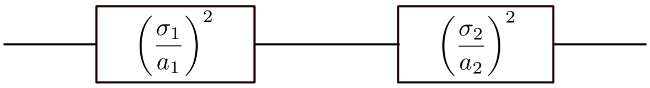

Figure 3.9: Circuit model in the information flow picture, equivalent to 2 Gaussian error sources acting independently, where the ”current” is a flow of information.

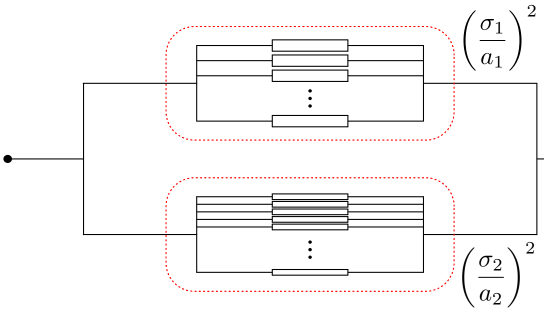

Figure 3.10: Circuit model in the error flow picture, equivalent to two Gaussian error sources acting independently, where the "current" is a flow of error. The sub circuits model how strong the influence of each error mode is on the logical error rate. In this case: $a^2 > b^2$.

For a certain amount of information flow, or R_{tot}, the relation between σ_1 and σ_2 becomes an elliptic expression $\left(\frac{\sigma_1}{a_1}\right)^2 + \left(\frac{\sigma_2}{a_2}\right)^2 = 1$. When we look for example at the case where $\sigma_1 = \sigma_2 = \sigma$, the expression becomes $\sigma^2 = \frac{1}{\frac{1}{a_1^2}+\frac{1}{a_2^2}}$. This is exactly the behavior we find in the simulations in previous sections. Note that this analogy only holds for fixed logical error rate. It breaks down when we study how a_n relates to P. This is not linear, as in an Ohmic circuit, but rather like discussed before and shown in fig. 3.8.

What can be concluded is that the error correction procedure does not introduce an interference between the noise sources. The noise sources independently affect the error rate. There is no consistent synergizing or canceling of the errors.

3.8 Surface-GKP as a particular case of generalized stabilizer codes.

A recent study [29] has shown theoretically that the concatenation of the GKP code with a stabilizer code, such as the rotated surface code, is a specific category within a more general family of multi-mode GKP codes. With our framework, we are in a position to corroborate this theory. The elliptic behavior observed in section 3.6 and treated in more detail in section 3.7, also appears for a single-mode GKP code. In fig. 3.11, the equivalent figure to 3.5 is shown with the fitted ellipsoid obeying:

$$\left(\frac{\sigma_0}{0.301}\right)^2 + \left(\frac{\sigma_1}{0.180}\right)^2 + \left(\frac{\sigma_2}{0.234}\right)^2 = 1. \tag{3.20}$$

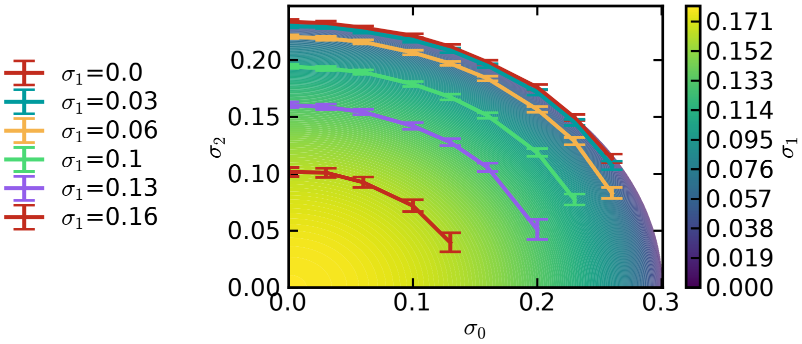

Figure 3.11: Ellipsoid fit to the contour at $P = 0.02$ for a single-mode GKP code. The curves are interpolated numerically obtained values of (σ_0,σ_1,σ_2) corresponding to $P = 0.02$. The numerical results are obtained by interpolating the logical error rate in σ_2 for each combination of (σ_1,σ_0) and finding the crossing with $P = 0.02$.

The vertices of the ellipse are larger now because the level $P = 0.02$ is above the threshold of the surface-GKP code. A naive way of seeing that they belong to the same family is by observing that a single-mode GKP code effectively constitutes a 1×1 surface code. A result that needs further study to fit with this interpretation, is the following. The error rate of a single mode GKP qubit obeys eq. (2.32) with an effective variance that is expected from section 3.3. Since the $\hat{q}$ variance is being corrected first, we look at step 2 and expect $\sqrt{2\sigma_{gkp}^2 + \frac{8}{3}\sigma_{sum}^2 + \sigma_{meas}^2 + 3\sigma_{photon}^2}$. With all variances equal, this evaluates to 2.944σ, which fits the results nicely in fig. 3.12. An equivalent transformation for multi-mode GKP codes is not obvious and would be an interesting subject of future enquiry.

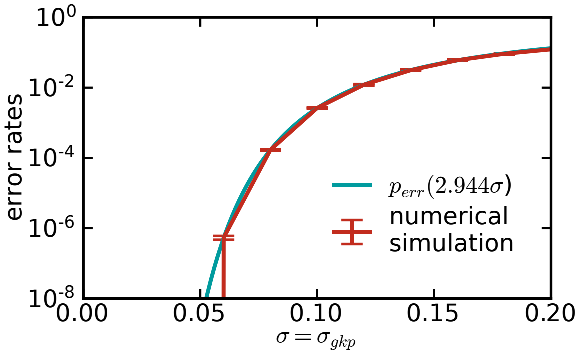

Figure 3.12: Error rate of a single-mode GKP qubit with all noise sources varied simultaneously, $\sigma = \sigma_{gkp}$. The blue curve represents eq. (2.32) with an effective variance of 2.944σ.

3.9 Influence of measurement efficiency on fault-tolerance threshold

In the interest of experimental work that might utilize this technique, we study the performance dependency on a quantity often listed for practical devices: the measurement efficiency η. This can be translated to the variance of the measurement noise as: $\sigma^2_{meas} = \frac{1-\eta}{2\eta}$ [106]. Fig. 3.13 shows the threshold σ_{gkp} as a function of the measurement threshold. All other noise sources are disabled in this simulation. The inclusion of analog information is again distinctly advantageous. This graph is in accordance with the elliptic behavior discussed in section 3.6, when one considers the re-scaling of the axis. One can also verify that the threshold at $\eta = 100\%$ agrees with the value reported in section 3.2. Finally, we can confirm that the expected value of η^* where the σ_{gkp} threshold disappears, corresponding to the threshold for σ_{meas} in section 3.5, agrees with the simulation in fig. 3.13. There is no threshold for the error correction code when the measurement inefficiency is under $\eta = 0.89$. Therefore, a high measurement efficiency is an instrumental part of a working error correction code. Currently, the micro-wave domain has not yet reached this level of efficiency, with reported values between $\eta = 0.4$ and $\eta = 0.75$ [107, 108, 109]. In the optical domain, this level would be attainable [110], but a major struggle there is the difficulty in obtaining the non-linearity needed for preparing GKP states. As of yet there have been no experimental showcases, but some simulations of proposed schemes have been promising [111, 112, 113]. The values reported here can be compared to previous results in ref. [106]. This is probably because they use a post-selection protocol called ”highly reliable measurement”. Apart from that, the fact that the Knill and Steane types of error-correction work differently could explain a difference.

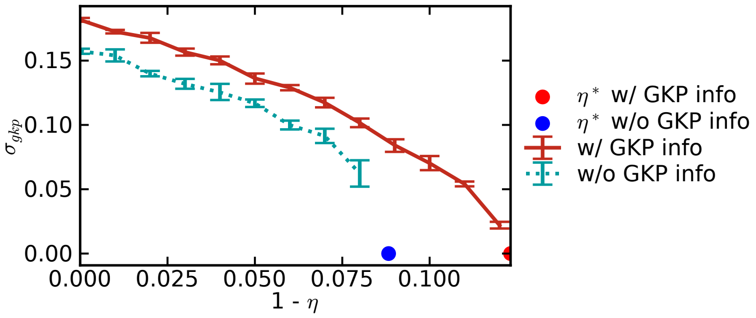

Figure 3.13: Influence of the measurement efficiency on the threshold GKP standard deviation. ”class 0” and ”class 2” errors are not considered in this simulation.

4

Conclusion and Outlook

4.1 Summary

Fault-tolerant error-correction is an important step on the road towards universal quantum computation. The aim of this thesis is to improve understanding the behavior of the rotated surface-GKP code by extensively exploring its performance when subjected to a realistic noise model. We obtain a more detailed notion of where the thresholds lie for the different noise sources. The study of the GKP threshold as function of measurement efficiency comes as a follow-up on the concern in ref. [95] about the need for improved detection schemes to cope with low measurement efficiency.

Comparison to variations on the surface-gkp code, based on the reported thresholds, is inconclusive as to whether the rotated surface-gkp code the best-performing one because the noise models, used in different studies, are not directly comparable. However, it has the advantage of being planar and relatively simple to implement.

By reproducing results from the original work we can verify our framework's validity. The extension on their framework lies in the more segregated expressions for expected noise on different qubits in the course of the procedure. These are reported in section 3.3.

In section 3.5, we find that the noise coming from $SUM_{j\to k}$ gates is most detrimental, as it has the lowest threshold at $\sigma^*_{sum} = 0.107$, and should be given special attention in physical implementations, while measurement noise is relatively the most tolerable with the highest threshold at $\sigma^*_{meas} = 0.107$. Noise coming from the overlap in realistic GKP states and that incurred while idling are comparable, with the latter being slightly more pervasive: $\sigma^*_{photon} = 0.162$ and $\sigma^*_{gkp} = 0.182$.

Classifying the noise sources into three categories in section 3.6, we simulate a surface corresponding to a constant logical error $P = 0.02$, which behaves like an ellipsoid. This corroborates the understanding of error modes as parallel channels for error-flow, explained in section 3.7, and the concatenated surface-GKP code as a particular case of a more general CV stabilizer code, explained in section 3.8. In comparison to the DV surface code, this is rather surprising. The constant logical error surface in the DV case is concave while it is convex in the CV case. We also find that the CV scheme is more sensitive to class-1 noise affecting the ancillary qubits than to class-0 noise affecting the data qubit, while it is the other way around for the DV case. Finally, in section 3.9, we find that a measurement efficiency well above 90% is mandatory for achieving fault-tolerance with this scheme in a realistic environment. Even though the requirements on measurement efficiency and GKP state preparation, for realizing fault-tolerant computation with the rotated surface-GKP code, have not yet materialized in practice, it remains a promising contender in the race to fault-tolerance.

4.2 Concerns

From our work, it has become apparent that the simulation bottleneck can be localized at the classical decoding of the surface code syndrome measurements. This raises the concern that a classical circuit monitoring the error correction of a quantum memory might not be able to keep up with the speed of quantum processes. Although the inefficient implementation at least greatly exaggerates this effect, it is still imperative that we know this is not a problem in physical implementations. Especially when the distance of the surface code increases, the processing time increases accordingly, while the coherence time of the physical qubits remains the same. This leads to a upper bound on the size of the surface, which is an interesting figure to study.

4.3 Future research

In future work, an interesting study that could be undertaken based on the code framework in this thesis, with minimal adjustments, is to simulate a discrete surface code with continuous measurement information [114]. One can imagine replacing the Gaussian displacements with discrete displacements of $\sqrt{\pi}$ for all noise sources except the measurement noise. However, a careful reconsideration of the noise model will be necessary. It will be interesting to implement this code into a faster simulation language like Julia [115], to increase the surface size and look for the thresholds at $d \to \infty$. Extrapolating this idea, and maybe a bit prematurely, one could build specialized hardware to verify that the algorithm can indeed keep up with the speed of the quantum processes [116]. In this work, we have ignored the relationship between noise and the time it takes to perform a cycle of error correction. This is a common approach for simulations of quantum error correction schemes because we are more interested in comparing schemes than quantifying absolute performances. Engineering efforts or algorithm optimizations can always reduce the time needed to perform the procedure. This is because we cannot realistically take this time into account as we are dealing with a sub-optimal implementation and have no measure of how fast the noise would grow over time compared to the speed of the simulation. However, the time it takes for an error correction cycle to complete is essential to its performance because it directly determines the error probability it will have to deal with.

On a theoretical front, some merit might come from an analytical study that could explain the phenomenon of the ellipsoid behavior of the error rate surfaces. Maybe state-twirling is responsible for the fact that the noise sources act independently. Finally, a resource or feasibility estimate would be very interesting for the future.

Bibliography

[1] Kyungjoo Noh and Christopher Chamberland. “Fault-tolerant bosonic quantum error correction with the surface–Gottesman-Kitaev-Preskill code”. In: *Phys. Rev. A* 101 (1), p. 012316. doi: `10.1103/PhysRevA.101.012316`. url: `https://link.aps.org/doi/10.1103/PhysRevA.101.012316`.

[2] Jonathan Conrad, Jens Eisert, and Francesco Arzani. “Gottesman-Kitaev-Preskill codes: A lattice perspective”. In: *Quantum* 6 (Feb. 2022), p. 648. issn: 2521-327X. doi: `10.22331/q-2022-02-10-648`. url: `https://doi.org/10.22331/q-2022-02-10-648`.

[3] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. doi: `10.1017/CBO9780511976667`.

[4] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. doi: `10.1109/SFCS.1994.365700`.

[5] Xin Zhang et al. “Semiconductor quantum computation”. In: *National Science Review* 6.1 (Dec. 2018), pp. 32–54. issn: 2095-5138. doi: `10.1093/nsr/nwy153`. eprint: `https://academic.oup.com/nsr/article-pdf/6/1/32/38914896/nwy153.pdf`. url: `https://doi.org/10.1093/nsr/nwy153`.

[6] He-Liang Huang et al. “Superconducting Quantum Computing: A Review”. In: (2020). doi: `10.48550/ARXIV.2006.10433`. url: `https://arxiv.org/abs/2006.10433`.

[7] Jerry M. Chow et al. “Universal Quantum Gate Set Approaching Fault-Tolerant Thresholds with Superconducting Qubits”. In: *Phys. Rev. Lett.* 109 (6 2012), p. 060501. doi: `10.1103/PhysRevLett.109.060501`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.109.060501`.

[8] Yu Chen et al. “Qubit Architecture with High Coherence and Fast Tunable Coupling”. In: *Phys. Rev. Lett.* 113 (22 2014), p. 220502. doi: `10.1103/PhysRevLett.113.220502`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.113.220502`.

[9] G Wendin. “Quantum information processing with superconducting circuits: a review”. In: *Reports on Progress in Physics* 80.10 (2017), p. 106001. doi: `10.1088/1361-6633/aa7e1a`. url: `https://doi.org/10.1088/1361-6633/aa7e1a`.

[10] B.E. Kane. “Silicon-based Quantum Computation”. In: *Fortschritte der Physik* 48.9-11 (2000), pp. 1023–1041. doi: `10.1002/1521-3978(200009)48:9/11<1023::aid-prop1023>3.0.co;2-j`. url: `https://doi.org/10.1002%2F1521-3978%28200009%2948%3A9%2F11%3C1023%3A%3Aaid-prop1023%3E3.0.co%3B2-j`.

[11] Joost van der Heijden et al. “Readout and control of the spin-orbit states of two coupled acceptor atoms in a silicon transistor”. In: *Science Advances* 4 (Dec. 2018), eaat9199. doi: `10.1126/sciadv.aat9199`.

[12] Charles D. Hill et al. “A surface code quantum computer in silicon”. In: *Science Advances* 1.9 (2015), e1500707. doi: `10.1126/sciadv.1500707`. eprint: `https://www.science.org/doi/pdf/10.1126/sciadv.1500707`. url: `https://www.science.org/doi/abs/10.1126/sciadv.1500707`.

[13] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. 1997. doi: `10.48550/ARXIV.QUANT-PH/9705052`. url: `https://arxiv.org/abs/quant-ph/9705052`.

[14] P. T. Cochrane, G. J. Milburn, and W. J. Munro. “Macroscopically distinct quantum-superposition states as a bosonic code for amplitude damping”. In: *Physical Review A* 59.4 (1999), pp. 2631–2634. doi: `10.1103/physreva.59.2631`. url: `https://doi.org/10.1103%2Fphysreva.59.2631`.

[15] Marios H. Michael et al. “New Class of Quantum Error-Correcting Codes for a Bosonic Mode”. In: *Phys. Rev. X* 6 (3 2016), p. 031006. doi: `10.1103/PhysRevX.6.031006`. url: `https://link.aps.org/doi/10.1103/PhysRevX.6.031006`.

[16] Daniel Gottesman, Alexei Kitaev, and John Preskill. “Encoding a qubit in an oscillator”. In: *Physical Review A* 64.1 (2001). doi: `10.1103/physreva.64.012310`. url: `https://doi.org/10.1103/2Fphysreva.64.012310`.

[17] Lianxue Hu et al. “Quantum error correction and universal gate set operation on a binomial bosonic logical qubit”. In: *Nature Physics* 15 (May 2019). doi: `10.1038/s41567-018-0414-3`.

[18] Jean Etesse et al. “Experimental Generation of Squeezed Cat States with an Operation Allowing Iterative Growth”. In: *Phys. Rev. Lett.* 114 (19 2015), p. 193602. doi: `10.1103/PhysRevLett.114.193602`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.114.193602`.

[19] C. Flühmann et al. “Encoding a qubit in a trapped-ion mechanical oscillator”. In: *Nature* 566.7745 (2019), pp. 513–517. doi: `10.1038/s41586-019-0960-6`. url: `https://doi.org/10.1038%2Fs41586-019-0960-6`.

[20] Emanuel Knill and Raymond Laflamme. *Concatenated Quantum Codes*. 1996. doi: `10.48550/ARXIV.QUANT-PH/9608012`. url: `https://arxiv.org/abs/quant-ph/9608012`.

[21] Jiaxuan Zhang et al. “Quantum error correction with the color-Gottesman-Kitaev-Preskill code”. In: *Physical Review A* 104.6 (2021). issn: 2469-9934. doi: `10.1103/physreva.104.062434`. url: `http://dx.doi.org/10.1103/PhysRevA.104.062434`.

[22] Christophe Vuillot et al. “Quantum error correction with the toric Gottesman-Kitaev-Preskill code”. In: *Phys. Rev. A* 99 (3 2019), p. 032344. doi: `10.1103/PhysRevA.99.032344`. url: `https://link.aps.org/doi/10.1103/PhysRevA.99.032344`.

[23] Mikkel V. Larsen et al. “Fault-Tolerant Continuous-Variable Measurement-based Quantum Computation Architecture”. In: *PRX Quantum* 2.3 (2021). issn: 2691-3399. doi: `10.1103/prxquantum.2.030325`. url: `http://dx.doi.org/10.1103/PRXQuantum.2.030325`.

[24] Austin G. Fowler. “Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time”. In: (2013). doi: `10.48550/ARXIV.1307.1740`. url: `https://arxiv.org/abs/1307.1740`.

[25] Austin G. Fowler, Adam C. Whiteside, and Lloyd C. L. Hollenberg. “Towards Practical Classical Processing for the Surface Code”. In: *Physical Review Letters* 108.18 (2012). doi: `10.1103/physrevlett.108.180501`. url: `https://doi.org/10.1103%2Fphysrevlett.108.180501`.

[26] Jack Edmonds. “Paths, Trees, and Flowers”. In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467. doi: `10.4153/CJM-1965-045-4`.

[27] Jack Edmonds. "Maximum Matching and a Polyhedron with $0, 1$ Vertices". In: *J. of Res. the Nat. Bureau of Standards* 69 B (1965), pp. 125–130.

[28] Yang Wang. *Quantum Error Correction with the GKP Code and Concatenation with Stabilizer Codes*. 2019. doi: `10.48550/ARXIV.1908.00147`. url: `https://arxiv.org/abs/1908.00147`.

[29] Jonathan Conrad. "Twirling and Hamiltonian engineering via dynamical decoupling for Gottesman-Kitaev-Preskill quantum computing". In: *Physical Review A* 103.2 (2021). doi: `10.1103/physreva.103.022404`. url: `https://doi.org/10.1103%2Fphysreva.103.022404`.

[30] R. W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. doi: `10.1002/j.1538-7305.1950.tb00463.x`.

[31] Alex M. Andrew. "INFORMATION THEORY, INFERENCE, AND LEARNING ALGORITHMS, by David J. C. MacKay, Cambridge University Press, Cambridge, 2003, hardback, xii 628 pp., ISBN 0-521-64298-1 (£30.00)". In: *Robotica* 22.3 (2004), pp. 348–349. doi: `10.1017/S026357470426043X`.

[32] Nikolas P. Breuckmann and Jens Niklas Eberhardt. "Quantum Low-Density Parity-Check Codes". In: *PRX Quantum* 2.4 (2021). doi: `10.1103/prxquantum.2.040101`. url: `https://doi.org/10.1103%2Fprxquantum.2.040101`.

[33] Mark M. Wilde. "Preface to the Second Edition". In: *Quantum Information Theory*. Cambridge University Press, pp. xi–xii. doi: `10.1017/9781316809976.001`. url: `https://doi.org/10.1017%2F9781316809976.001`.

[34] Caleb Hillier and Vipin Balyan. "Error Detection and Correction On-Board Nanosatellites Using Hamming Codes". In: *J. Electr. Comput. Eng.* 2019 (2019), 3905094:1–3905094:15.

[35] David Fiala et al. "Detection and correction of silent data corruption for large-scale high-performance computing". In: Jan. 2011, pp. 47–48. doi: `10.1109/SC.2012.49`.

[36] G. David Forney et al. "Efficient Modulation for Band-Limited Channels". In: *IEEE J. Sel. Areas Commun.* 2 (1984), pp. 632–647.

[37] Pradeep M. Shah, Prakash D. Vyavahare, and Anjana Jain. "Modern error correcting codes for 4G and beyond: Turbo codes and LDPC codes". In: *2015 Radio and Antenna Days of the Indian Ocean (RADIO)*. 2015, pp. 1–2. doi: `10.1109/RADIO.2015.7323369`.

[38] John von Neumann. “Probabilistic Logic and the Synthesis of Reliable Organisms from Unreliable Components”. In: 1956.

[39] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: 299.5886 (Oct. 1982), pp. 802–803. doi: `10.1038/299802a0`.

[40] Emanuel Knill and Raymond Laflamme. “Theory of quantum error-correcting codes”. In: *Phys. Rev. A* 55 (2 1997), pp. 900–911. doi: `10.1103/PhysRevA.55.900`. url: `https://link.aps.org/doi/10.1103/PhysRevA.55.900`.

[41] Charles H. Bennett et al. “Mixed-state entanglement and quantum error correction”. In: *Phys. Rev. A* 54 (5 1996), pp. 3824–3851. doi: `10.1103/PhysRevA.54.3824`. url: `https://link.aps.org/doi/10.1103/PhysRevA.54.3824`.

[42] A. R. Calderbank et al. *Quantum Error Correction via Codes over GF(4)*. 1996. doi: `10.48550/ARXIV.QUANT-PH/9608006`. url: `https://arxiv.org/abs/quant-ph/9608006`.

[43] Avanti Ketkar et al. *Nonbinary stabilizer codes over finite fields*. 2005. doi: `10.48550/ARXIV.QUANT-PH/0508070`. url: `https://arxiv.org/abs/quant-ph/0508070`.

[44] Graeme Smith and John A. Smolin. “Degenerate Quantum Codes for Pauli Channels”. In: *Physical Review Letters* 98.3 (2007). doi: `10.1103/physrevlett.98.030501`. url: `https://doi.org/10.1103%2Fphysrevlett.98.030501`.

[45] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (1996), pp. 1098–1105. doi: `10.1103/physreva.54.1098`. url: `https://doi.org/10.1103%2Fphysreva.54.1098`.

[46] “Multiple-particle interference and quantum error correction”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. doi: `10.1098/rspa.1996.0136`. url: `https://doi.org/10.1098%2Frspa.1996.0136`.

[47] A Yu Kitaev. “Quantum computations: algorithms and error correction”. In: *Russian Mathematical Surveys* 52.6 (1997), pp. 1191–1249. doi: `10.1070/rm1997v052n06abeh002155`. url: `https://doi.org/10.1070/rm1997v052n06abeh002155`.

[48] A. Yu. Kitaev. “Quantum Error Correction with Imperfect Gates”. In: *Quantum Communication, Computing, and Measurement*. Ed. by O. Hirota, A. S. Holevo, and C. M. Caves. Boston, MA: Springer US, 1997, pp. 181–188. isbn: 978-1-4615-5923-8. doi: `10.1007/978-1-4615-5923-8_19`. url: `https://doi.org/10.1007/978-1-4615-5923-8_19`.

[49] A.Yu. Kitaev. “Fault-tolerant quantum computation by anyons”. In: *Annals of Physics* 303.1 (2003), pp. 2–30. doi: `10.1016/s0003-4916(02)00018-0`. url: `https://doi.org/10.1016%2Fs0003-4916%2802%2900018-0`.

[50] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (2012). issn: 1094-1622. doi: `10.1103/physreva.86.032324`. url: `http://dx.doi.org/10.1103/PhysRevA.86.032324`.

[51] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (2002), pp. 4452–4505. doi: `10.1063/1.1499754`. eprint: `https://doi.org/10.1063/1.1499754`. url: `https://doi.org/10.1063/1.1499754`.

[52] S. B. Bravyi and A. Yu. Kitaev. *Quantum codes on a lattice with boundary*. 1998. doi: `10.48550/ARXIV.QUANT-PH/9811052`. url: `https://arxiv.org/abs/quant-ph/9811052`.

[53] Armands Strikis, Simon C. Benjamin, and Benjamin J. Brown. *Quantum computing is scalable on a planar array of qubits with fabrication defects*. 2021. doi: `10.48550/ARXIV.2111.06432`. url: `https://arxiv.org/abs/2111.06432`.

[54] Youwei Zhao et al. *Realization of an Error-Correcting Surface Code with Superconducting Qubits*. 2021. doi: `10.48550/ARXIV.2112.13505`. url: `https://arxiv.org/abs/2112.13505`.

[55] Sebastian Krinner et al. *Realizing Repeated Quantum Error Correction in a Distance-Three Surface Code*. 2021. doi: `10.48550/ARXIV.2112.03708`. url: `https://arxiv.org/abs/2112.03708`.

[56] David K. Tuckett et al. “Tailoring Surface Codes for Highly Biased Noise”. In: *Physical Review X* 9.4 (2019). doi: `10.1103/physrevx.9.041031`. url: `https://doi.org/10.1103%2Fphysrevx.9.041031`.

[57] Austin G. Fowler. “Low-overhead surface code logical Hadamard”. In: (2012). doi: `10.48550/ARXIV.1202.2639`. url: `https://arxiv.org/abs/1202.2639`.

[58] Clare Horsman et al. “Surface code quantum computing by lattice surgery”. In: *New Journal of Physics* 14.12 (2012), p. 123011. doi: `10.1088/1367-2630/14/12/123011`. url: `https://doi.org/10.1088/1367-2630/14/12/123011`.

[59] Nicolas Delfosse and Gilles Zé mor. “Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel”. In: *Physical Review Research* 2.3 (2020). doi: `10.1103/physrevresearch.2.033042`. url: `https://doi.org/10.1103%2Fphysrevresearch.2.033042`.

[60] Sergey Bravyi, Martin Suchara, and Alexander Vargo. “Efficient algorithms for maximum likelihood decoding in the surface code”. In: *Physical Review A* 90.3 (2014). doi: `10.1103/physreva.90.032326`. url: `https://doi.org/10.1103%2Fphysreva.90.032326`.

[61] Yu Tomita and Krysta M. Svore. “Low-distance surface codes under realistic quantum noise”. In: *Phys. Rev. A* 90 (6 2014), p. 062320. doi: `10.1103/PhysRevA.90.062320`. url: `https://link.aps.org/doi/10.1103/PhysRevA.90.062320`.

[62] Hendrik Poulsen Nautrup et al. “Optimizing Quantum Error Correction Codes with Reinforcement Learning”. In: *Quantum* 3 (2019), p. 215. doi: `10.22331/q-2019-12-16-215`. url: `https://doi.org/10.22331%2Fq-2019-12-16-215`.

[63] Debasmita Bhoumik et al. *Efficient Decoding of Surface Code Syndromes for Error Correction in Quantum Computing*. 2021. doi: `10.48550/ARXIV.2110.10896`. url: `https://arxiv.org/abs/2110.10896`.

[64] Savvas Varsamopoulos, Ben Criger, and Koen Bertels. “Decoding small surface codes with feedforward neural networks”. In: *Quantum Science and Technology* 3.1 (2017), p. 015004. doi: `10.1088/2058-9565/aa955a`. url: `https://doi.org/10.1088%2F2058-9565%2Faa955a`.

[65] Edsger W Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[66] jarrodmillman. *Networkx 2.6.3*. 2021. url: `https://github.com/networkx` (visited on 04/05/2022).

[67] Zvi Galil. “Efficient Algorithms for Finding Maximum Matching in Graphs”. In: *ACM Comput. Surv.* 18.1 (1986), pp. 23–38. issn: 0360-0300. doi: `10.1145/6462.6502`. url: `https://doi.org/10.1145/6462.6502`.

[68] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (4 1995), R2493–R2496. doi: `10.1103/PhysRevA.52.R2493`. url: `https://link.aps.org/doi/10.1103/PhysRevA.52.R2493`.

[69] Peter W. Shor. *Fault-tolerant quantum computation*. 1996. doi: `10.48550/ARXIV.QUANT-PH/9605011`. url: `https://arxiv.org/abs/quant-ph/9605011`.

[70] Barbara M. Terhal. “Quantum error correction for quantum memories”. In: *Reviews of Modern Physics* 87.2 (2015), pp. 307–346. doi: `10.1103/revmodphys.87.307`. url: `https://doi.org/10.1103%2Frevmodphys.87.307`.

[71] Daniel Gottesman. *An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation*. 2009. doi: `10.48550/ARXIV.0904.2557`. url: `https://arxiv.org/abs/0904.2557`.

[72] P. Vikstål. “Continuous-variable quantum annealing with superconducting circuits.” MA thesis. Chalmers, 2018.

[73] G Wendin. “Quantum information processing with superconducting circuits: a review”. In: *Reports on Progress in Physics* 80.10 (2017), p. 106001. doi: `10.1088/1361-6633/aa7e1a`. url: `https://doi.org/10.1088%2F1361-6633%2Faa7e1a`.

[74] Sergei Slussarenko and Geoff J. Pryde. “Photonic quantum information processing: A concise review”. In: *Applied Physics Reviews* 6.4 (2019), p. 041303. doi: `10.1063/1.5115814`. eprint: `https://doi.org/10.1063/1.5115814`. url: `https://doi.org/10.1063/1.5115814`.

[75] Seth Lloyd and Samuel L. Braunstein. “Quantum Computation over Continuous Variables”. In: *Phys. Rev. Lett.* 82 (8 1999), pp. 1784–1787. doi: `10.1103/PhysRevLett.82.1784`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.82.1784`.

[76] Samuel L. Braunstein and Peter van Loock. “Quantum information with continuous variables”. In: *Rev. Mod. Phys.* 77 (2 2005), pp. 513–577. doi: `10.1103/RevModPhys.77.513`. url: `https://link.aps.org/doi/10.1103/RevModPhys.77.513`.

[77] Nicolas C. Menicucci et al. “Universal Quantum Computation with Continuous-Variable Cluster States”. In: *Phys. Rev. Lett.* 97 (11 2006), p. 110501. doi: `10.1103/PhysRevLett.97.110501`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.97.110501`.

[78] Francesco Lenzini et al. “Integrated photonic platform for quantum information with continuous variables”. In: *Science Advances* 4.12 (2018), eaat9331. doi: `10.1126/sciadv.aat9331`. eprint: `https://www.science.org/doi/pdf/10.1126/sciadv.aat9331`. url: `https://www.science.org/doi/abs/10.1126/sciadv.aat9331`.

[79] Christian Weedbrook et al. “Gaussian quantum information”. In: *Reviews of Modern Physics* 84.2 (2012), pp. 621–669. doi: `10.1103/revmodphys.84.621`. url: `https://doi.org/10.1103%2Frevmodphys.84.621`.

[80] Julien Nisetek, familyi=v., given=Jaromír, giveni=J., „ and Nicolas J. Cerf. “No-Go Theorem for Gaussian Quantum Error Correction”. In: *Phys. Rev. Lett.* 102 (12 2009), p. 120501. doi: `10.1103/PhysRevLett.102.120501`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.102.120501`.

[81] Christopher Chamberland et al. “Building a Fault-Tolerant Quantum Computer Using Concatenated Cat Codes”. In: *PRX Quantum* 3.1 (2022). doi: `10.1103/prxquantum.3.010329`. url: `https://doi.org/10.1103%2Fprxquantum.3.010329`.

[82] Victor V. Albert et al. “Performance and structure of single-mode bosonic codes”. In: *Physical Review A* 97.3 (2018). doi: `10.1103/physreva.97.032346`. url: `https://doi.org/10.1103%2Fphysreva.97.032346`.

[83] Jacob Hastrup et al. “Unsuitability of cubic phase gates for non-Clifford operations on Gottesman-Kitaev-Preskill states”. In: *Phys. Rev. A* 103 (3 2021), p. 032409. doi: `10.1103/PhysRevA.103.032409`. url: `https://link.aps.org/doi/10.1103/PhysRevA.103.032409`.

[84] Ilan Tzitrin et al. “Fault-Tolerant Quantum Computation with Static Linear Optics”. In: *PRX Quantum* 2 (4 2021), p. 040353. doi: `10.1103/PRXQuantum.2.040353`. url: `https://link.aps.org/doi/10.1103/PRXQuantum.2.040353`.

[85] B. M. Terhal and D. Weigand. “Encoding a qubit into a cavity mode in circuit QED using phase estimation”. In: *Phys. Rev. A* 93 (1 2016), p. 012315. doi: `10.1103/PhysRevA.93.012315`. url: `https://link.aps.org/doi/10.1103/PhysRevA.93.012315`.

[86] Yunong Shi, Christopher Chamberland, and Andrew Cross. “Fault-tolerant preparation of approximate GKP states”. In: *New Journal of Physics* 21.9 (2019), p. 093007. doi: `10.1088/1367-2630/ab3a62`. url: `https://doi.org/10.1088/1367-2630/ab3a62`.

[87] Giacomo Pantaleoni, Ben Q. Baragiola, and Nicolas C. Menicucci. “Modular Bosonic Subsystem Codes”. In: *Physical Review Letters* 125.4 (2020). doi: `10.1103/physrevlett.125.040501`. url: `https://doi.org/10.1103%2Fphysrevlett.125.040501`.

[88] Ilan Tzitrin et al. “Progress towards practical qubit computation using approximate Gottesman-Kitaev-Preskill codes”. In: *Phys. Rev. A* 101 (3 2020), p. 032315. doi: `10.1103/PhysRevA.101.032315`. url: `https://link.aps.org/doi/10.1103/PhysRevA.101.032315`.

[89] Takaya Matsuura, Hayata Yamasaki, and Masato Koashi. “Equivalence of approximate Gottesman-Kitaev-Preskill codes”. In: *Physical Review A* 102.3 (2020). doi: `10.1103/physreva.102.032408`. url: `https://doi.org/10.1103%2Fphysreva.102.032408`.

[90] “Logical error rate in the Pauli twirling approximation”. In: *Scientific Reports* 5 (1 2015). doi: `10.1038/srep14670`. url: `https://doi.org/10.1038/srep14670`.

[91] Joseph Emerson et al. “Symmetrized Characterization of Noisy Quantum Processes”. In: *Science* 317.5846 (2007), pp. 1893–1896. issn: 1095-9203. doi: `10.1126/science.1145699`. url: `http://dx.doi.org/10.1126/science.1145699`.

[92] Amara Katabarwa. *A dynamical interpretation of the Pauli Twirling Approximation and Quantum Error Correction*. 2017. doi: `10.48550/ARXIV.1701.03708`. url: `https://arxiv.org/abs/1701.03708`.

[93] Andrew S. Darmawan and David Poulin. “Tensor-Network Simulations of the Surface Code under Realistic Noise”. In: *Physical Review Letters* 119.4 (2017). doi: `10.1103/physrevlett.119.040502`. url: `https://doi.org/10.1103%2Fphysrevlett.119.040502`.

[94] Mauricio Gutiérrez and Kenneth R. Brown. “Comparison of a quantum error-correction threshold for exact and approximate errors”. In: *Phys. Rev. A* 91 (2 2015), p. 022335. doi: `10.1103/PhysRevA.91.022335`. url: `https://link.aps.org/doi/10.1103/PhysRevA.91.022335`.

[95] Timo Hillmann et al. *Performance of teleportation-based error correction circuits for bosonic codes with noisy measurements*. 2021. arXiv: `2108.01009 [quant-ph]`.

[96] Kosuke Fukui et al. “High-Threshold Fault-Tolerant Quantum Computation with Analog Quantum Error Correction”. In: *Phys. Rev. X* 8 (2 2018), p. 021054. doi: `10.1103/PhysRevX.8.021054`. url: `https://link.aps.org/doi/10.1103/PhysRevX.8.021054`.

[97] Kyungjoo Noh, Christopher Chamberland, and Fernando G.S.L. Brandão. “Low-Overhead Fault-Tolerant Quantum Error Correction with the Surface-GKP Code”. In: *PRX Quantum* 3 (1 2022), p. 010315. doi: `10.1103/PRXQuantum.3.010315`. url: `https://link.aps.org/doi/10.1103/PRXQuantum.3.010315`.

[98] Lisa Hänggli, Margret Heinze, and Robert König. “Enhanced noise resilience of the surface–Gottesman-Kitaev-Preskill code via designed bias”. In: *Phys. Rev. A* 102 (5 2020), p. 052408. doi: `10.1103/PhysRevA.102.052408`. url: `https://link.aps.org/doi/10.1103/PhysRevA.102.052408`.

[99] E Knill. “Quantum computing with realistically noisy devices”. In: *Nature* 434.7029 (2005), pp. 39—44. issn: 0028-0836. doi: `10.1038/nature03350`. url: `https://doi.org/10.1038/nature03350`.

[100] David P. DiVincenzo and Panos Aliferis. “Effective Fault-Tolerant Quantum Computation with Slow Measurements”. In: *Phys. Rev. Lett.* 98 (2 2007), p. 020501. doi: `10.1103/PhysRevLett.98.020501`. url: `https://link.aps.org/doi/10.1103/PhysRevLett.98.020501`.

[101] Nicolas C. Menicucci. “Fault-Tolerant Measurement-Based Quantum Computing with Continuous-Variable Cluster States”. In: *Physical Review Letters* 112.12 (2014). doi: `10.1103/physrevlett.112.120504`. url: `https://doi.org/10.1103%2Fphysrevlett.112.120504`.

[102] Kosuke Fukui, Rafael N. Alexander, and Peter van Loock. “All-optical long-distance quantum communication with Gottesman-Kitaev-Preskill qubits”. In: *Physical Review Research* 3.3 (2021). doi: `10.1103/physrevresearch.3.033118`. url: `https://doi.org/10.1103%2Fphysrevresearch.3.033118`.

[103] C. Ryan-Anderson et al. “Realization of Real-Time Fault-Tolerant Quantum Error Correction”. In: *Physical Review X* 11 (Dec. 2021). doi: `10.1103/PhysRevX.11.041058`.

[104] B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathemat- ics, Philadelphia, 1982. doi: `10.1137/1.9781611970319.fm`.

[105] Kyungjoo Noh. “Quantum Computation and Communication in Bosonic Systems”. In: (2021). doi: `10.48550/ARXIV.2103.09445`. url: `https://arxiv.org/abs/2103.09445`.

[106] Kosuke Fukui. *High-threshold fault-tolerant quantum computation with the GKP qubit and realistically noisy devices*. 2019. doi: `10.48550/ARXIV.1906.09767`. url: `https://arxiv.org/abs/1906.09767`.

[107] Wolfgang Pfaff et al. “Controlled release of multiphoton quantum states from a microwave cavity memory”. In: *Nature Physics* 13.9 (Sept. 2017), pp. 882–887. doi: `10.1038/nphys4143`. arXiv: `1612.05238 [quant-ph]`.

[108] Leigh S. Martin et al. “Implementation of a canonical phase measurement with quantum feedback”. In: *Nature Physics* 16.10 (2020), pp. 1046–1049. doi: `10.1038/s41567-020-0939-0`. url: `https://doi.org/10.1038%2Fs41567-020-0939-0`.

[109] T. Walter et al. “Rapid High-Fidelity Single-Shot Dispersive Readout of Superconducting Qubits”. In: *Phys. Rev. Applied* 7 (5 2017), p. 054020. doi: `10.1103/PhysRevApplied.7.054020`. url: `https://link.aps.org/doi/10.1103/PhysRevApplied.7.054020`.

[110] H. Hansen et al. “Ultrasensitive pulsed, balanced homodyne detector: application to time-domain quantum measurements”. In: *Optics Letters* 26.21 (2001), p. 1714. doi: `10.1364/ol.26.001714`. url: `https://doi.org/10.1364%2Fol.26.001714`.

[111] H. M. Vasconcelos, L. Sanz, and S. Glancy. “All-optical generation of states for “Encoding a qubit in an oscillator””. In: *Optics Letters* 35.19 (2010), p. 3261. doi: `10.1364/ol.35.003261`. url: `https://doi.org/10.1364%2Fol.35.003261`.

[112] Kosuke Fukui et al. “Generating the Gottesman-Kitaev-Preskill qubit using a cross-Kerr interaction between squeezed light and Fock states in optics”. In: *Phys. Rev. A* 105 (2 2022), p. 022436. doi: `10.1103/PhysRevA.105.022436`. url: `https://link.aps.org/doi/10.1103/PhysRevA.105.022436`.

[113] Jacob Hastrup and Ulrik L. Andersen. *Generation of optical Gottesman-Kitaev-Preskill states with cavity QED*. 2021. doi: `10.48550/ARXIV.2104.07981`. url: `https://arxiv.org/abs/2104.07981`.

[114] Christopher A. Pattison et al. *Improved quantum error correction using soft information*. 2021. doi: `10.48550/ARXIV.2107.13589`. url: `https://arxiv.org/abs/2107.13589`.

[115] Jeff Bezanson et al. *Julia: A Fast Dynamic Language for Technical Computing*. 2012. doi: `10.48550/ARXIV.1209.5145`. url: `https://arxiv.org/abs/1209.5145`.

[116] Poulami Das et al. *A Scalable Decoder Micro-architecture for Fault-Tolerant Quantum Computing*. 2020. doi: `10.48550/ARXIV.2001.06598`. url: `https://arxiv.org/abs/2001.06598`.