Surface Code Decoding under Spatially Correlated Errors

Yorgos Sotiropoulos*

Supervisors: Ng Hui Khoon†, Terhal Barbara‡
Collaborators: Chai Jing Hao§, Mark Myers II¶

February, 2022

Contents

Abstract

Surface codes can be improved by taking into account spatial correlations while decoding. We showcase this by modifying the weights of the Minimum Weight Perfect Matching algorithm or adding new connections with modified weights. We employ two different noise models, both variants of the Pauli noise model: one that is spatially inhomogeneous and one homogeneous with the added possibility of correlated noise model among neighbouring data qubits.

1 Introduction

One of the greatest challenges in the era of Quantum Computers is achieving fault-tolerance, which is the ability to complete tasks even when the underlying architecture is prone to having errors. Suppression of the errors can be improved by upgrading the hardware and by employing Quantum Error Correction.

Quantum error correction is the process of protecting quantum information. [8] This is achieved by encoding a collection of physical qubits into fewer or a single logical qubit, thus minimizing the effect of errors at the expense of computational qubits. This is possible because the logical qubit will not undergo error even if some of the underlying physical qubits do so. There are many Quantum Error Correcting protocols that achieve that at different degrees and one of the most promising is the Surface Code.

The Surface Code[4] relies on an architecture that only allows for a two-dimensional qubit arrangement on a grid, with a limited amount of nearest neighbour connections. These "connections" refer to the ability of executing a CNOT gate in a single step, without the need for added SWAP gates. These protocols have the advantage that the underlying architecture is realistic for Quantum Processors whether it be based on Cavity or Circuit Quantum Electrodynamics.

*MSc Student, Delft University of Technology, undertaking this project as part of Applied Sciences MSc Honors Program
†Associate Professor at Yale-NUS College/ Center for Quantum Technologies Fellow, Singapore
‡Professor at QuTech, Delft University of Technology, Netherlands
§Postdoctoral Research Fellow, Center for Quantum Technologies, Singapore
¶PhD Student, Center for Quantum Technologies, Singapore

The decoding process consists of extracting syndromes and determining the errors that have occured based on the syndromes. The standard method of decoding which is based on Edmond's Minimum Weight Perfect Matching (MWPM) algorithm [3] suffers loss of performance in the presence of spatial or temporal correlations since the underlying assumption is that every elementary error event is independent and equiprobable.

In this work we explore the possibility of enhancing the performance of the MWPM by tweaking weights appropriately depending on the underlying noise model. We introduce two variants of the of the single qubit Paul[7] noise model: one that is spatially dependent and one that is spatially correlated. We benchmark the performance of the MWPM against a weighted version of it (WMWPM).

2 Background on the Surface Code

A significant parameter for the Surface Code is the *distance* d which is the size of the array encoding a single logical qubit.[4] A code with distance d can always correct $\frac{d-1}{2}$ arbitrary errors with perfect stabilizer measurements.[5] The number of physical qubits needed varies based on the way qubits are arranged and connected.

In this work, we shall be assuming that the physical qubits are arranged on a square grid where the connections are the edges and the physical qubits are the vertices. This means that $(2d-1)^2$ qubits are needed, approximately half of which ($\lceil \frac{(2d-1)^2}{2} \rceil$) are used as data qubits and the rest are used as auxiliary qubits. The data qubits are used to encode the logical qubit while the auxiliary qubits are used to measure the necessary stabilizers[8] and thus detect any potential noise induced errors.

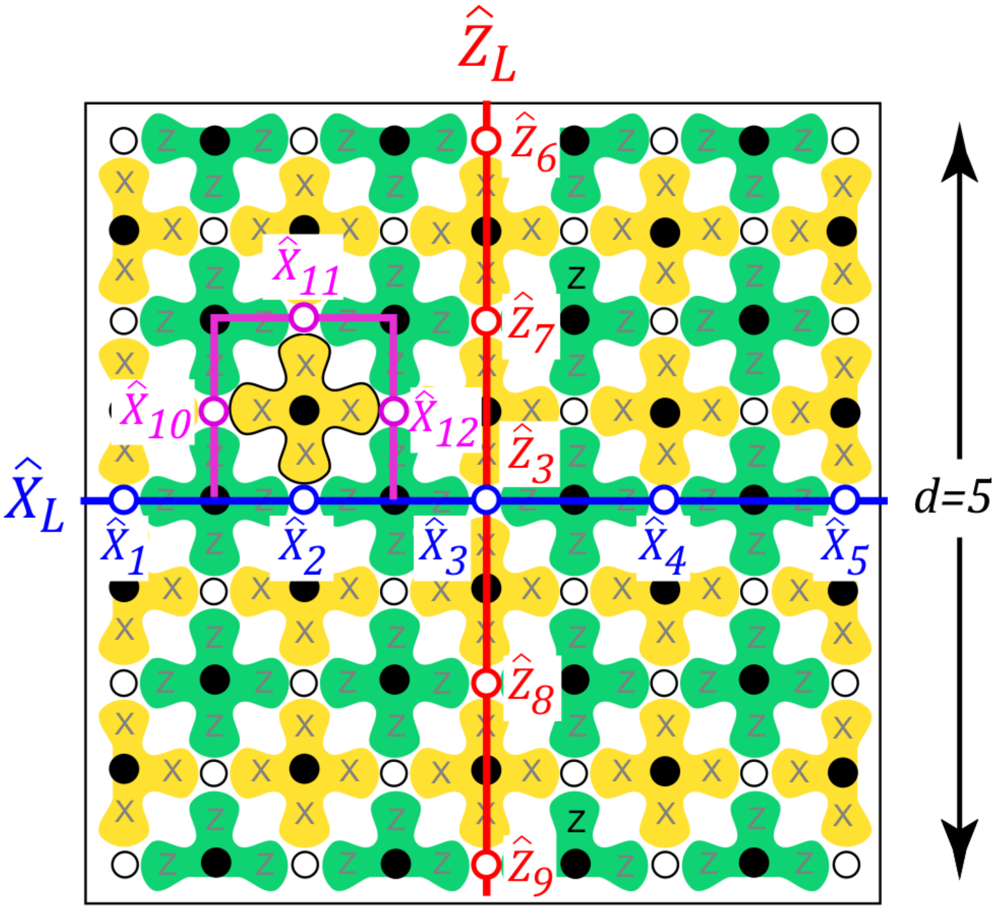


Figure 1: Surface code logical qubit encoding for distance 5. Data qubits are denoted as white circles and auxiliary ones as dark circles. Some logical X and Z operations are also visible. Note that $\mathbf{X_{L1}} = X_1X_2X_3X_4X_5$ and $\mathbf{X_{L2}} = X_1X_{10}X_{11}X_{12X_3}X_4X_5$ are equivalent logical operations since $\mathbf{X_{L1}} = \mathbf{X_{L1}}X_{10}X_{11}X_{12}X_2$ with $X_{10}X_{11}X_{12}X_2$ being a stabilizer. Figure from [4].

The stabilizers are of two types: plaquettes and vertices.

- Plaquette stabilizers are of the form $Z_iZ_jZ_kZ_l$ and all of them collectively are used to detect X errors.

- Vertex stabilizers are of the form $X_i X_j X_k X_l$ and are similarly used to detect Z errors.

Y errors that can be decomposed into X and Z errors will be detected by both.

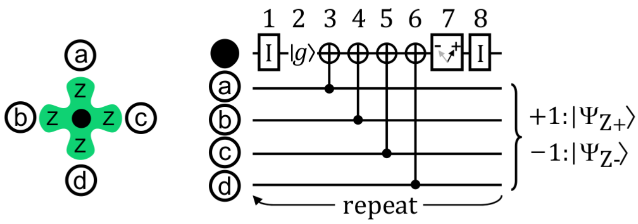


Figure 2: Plaquette stabilizer circuit from [4]

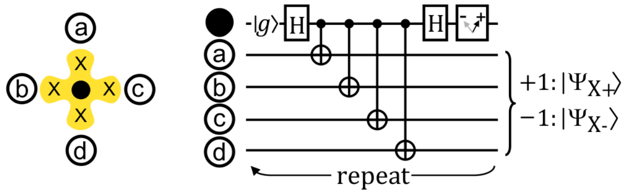


Figure 3: Vertex stabilizer circuit from [4]

A logical X or logical Z operation consists of a strip of gates across the lattice like in figure 1. These operations commute with the stabilizers, thus they are undetected and cannot be be written as a product of stabilizers.[4] A logical error similarly will consist of errors and/or recovery gates addressing misidentified errors that will form a chain across the lattice.

Decoding/Recovery: After performing the stabilizer measurements, we can separate the syndromes into two lattices: one for the plaquettes and one for the vertices. If an error occurs on the data qubits, the neighbouring auxiliary qubits that can detect the change in the stabilizer that they are measuring, will register two defects in the appropriate lattice. Similarly, if a continuous line of the same type of error occurs, two defects will be registered at the ends of the line. Thus, the syndrome lattice will be comprised of defects that need to be paired up among themselves in a way that would correspond to the most likely underlying errors. See Figure 4 for visualization.

For example, the plaquette syndrome lattice is a square lattice where the vertices are the Z-stabilizer auxiliary qubits and we can imagine the data qubits lying at the edges. X errors will cause defects to appear in the lattice that need to be paired up for the recovery. This can be achieved in general by employing Edmond's Minimum Weight Perfect Matching algorithm [3] and weighing the connection of each defect to the every other one with their Manhattan distance[1].[7] To address boundary cases, we allow the nodes to be connected with the boundary by assuming some ghost nodes.[6]

Worth noting also is that we do not need to exactly select the correct recovery but rather the correct homology.[6] To showcase this, let us imagine on Figure 1 that no error happens apart from X_2 and X_{12}. It is trivial to see how the defects on the plaquette stabilizers shall be paired up, generalising to two dimensions Figure 4. However, this error syndrome could also have come from the errors X_{10} and X_{11}. The good news is that we do not need to differentiate between the two error events due to them forming a stabilizer. If we choose the recovery $X_2 X_{12}$ then we indeed have undone the error, while if we choose the recovery $X_{10} X_{11}$ then we have effectively acted trivially on the logical state by applying the stabilizer $X_{10} X_{11} X_{12} X_2$. In both case, the logical state shall remain intact.

Although in this work we shall not be treading outside perfect stabilizer measurements, it is worth noting that when everything is allowed to be erroneous, including the stabilizer measurements, a different approach is needed. In these cases, the surface code is performed over and over again without decoding but just bookkeeping the extracted syndromes. In the end of the process, the decoding process is performed by "stitching" together the syndrome lattices into a three dimensional syndrome volume.[6]

[1]Manhattan distance between two vertices on a square grid is the distance when following only the grid lines.

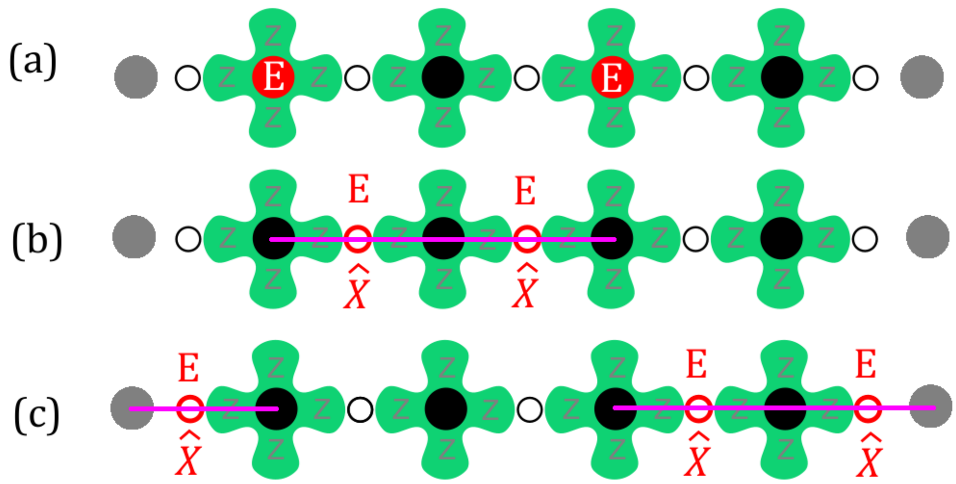


Figure 4: Syndrome decoding of a an isolated row of plaquette stabilizers of a distance 5 array, with the grey dots being ghost nodes that we assume for our convenience.
(a) Syndrome which can be attributed to underlying errors that could come up as the result of errors depicted in (b) or (c). Note that when two X-errors occur next to a plaquette auxiliary qubit, its stabilizer does not register a flip.
To decode (b) successfully, the defects have to be paired with each other.
To decode (c) successfully, the defects have to be paired with the boundary.
If all single error events are equiprobable, (b) should be assumed, since it is more likely than (c). Misidentification of the underlying error will lead to a decoding that combined with the errors will produce a logical X error. Adaptation of figure from [4]

3 Code discussion

The code that we shall be using is designed for fault-tolerance benchmarking of the Surface Code. In particular the simulation is treated as a stabilizer circuit [1, 5], which means that it only consists of CNOT, Hadamard, Phase gates[2] and measurements. The full circuit is simulated but because every operation is part of the Clifford Group, the simulation can be optimized in a way that it does not come with a large computational overhead.[1]

This has the advantage that errors can be modelled into every step of the surface code, including noise on auxiliary qubits. Moreover, a Surface Code cycle is comprised of many gates that could be erroneous, so assuming that the errors happen only in between two consecutive cylces is not realistic. In fact, errors at different points within a Surface Code cycle will produce various non trivial syndromes to arise from a single error event.[6] These syndromes often span more than one time-slice of the three dimensional syndrome lattice and are much more realistic than simple error correcting codes.

In our work we shall be using this code and handicapping it to simulate Error Correcting Codes in the *code capacity* setting, i.e. with perfect stabilizer measurements and one single time-slice. This means that we will not utilize the full potential of the code while simultaneously making the collection of data more computationally expensive. However, this means that expanding the results below to a fault-tolerant implementation with multiple time-slices becomes more feasible in the future.

[2]Phase gates refer to the unitary $P = diag(1, e^{i\frac{\pi}{2}})$

4 Improving decoding using spatial information of the noise

When employing the MWPM algorithm, we have a static square lattice that every edge is weighted equally and thus, connecting any two defects comes with a weight equal to the Manhattan distance, which is the shortest path distance. In our endeavor we shall be tweaking this geometry and the edges will not be equally weighted or we will allow for shortcut-connections among vertices.

In this section we shall explore the possibilities of enhancing the performance of the MWPM algorithm. Before introducing the noise models that we shall be benchmarking against, we first need to discuss a bit about how we can use the spatial information when assigning the weights.

4.1 Selecting a function for the WMWPM

The weights based on some function f which has as an input the error probability that would cause such an underlying event.

Let two defect nodes A, B be connected two ways:

- a path which passes through a third node C
- a path which directly connects them

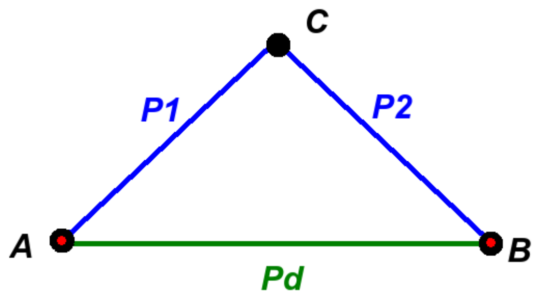


Figure 5: The two defect nodes A,B are connected directly due to some error event Pd and indirectly due to the simultaneous realisation of $P1$ and $P2$.

Let both of those paths be equivalent. This means that the underlying error event that would cause the A-B connection is equiprobable with the simultaneous realisation of the error events that would cause A-C and C-B connections. The probabilities of error in this case should satisfy:

$$p_d = p_1 p_2 \tag{1}$$

But also:

$$w_d = w_1 + w_2 \tag{2}$$

so that the MWPM algorithm does not discriminate between those two paths.

$$f(p_d) = f(p_1) + f(p_2) \tag{3}$$

We see that this can be satisfied by:

$$f(p) = \pm\, log(p) \tag{4}$$

Since we also need to favor paths with higher probability of the underlying event happening, those paths should have a lower weight allocated to them.

$$p_i > p_j \Rightarrow w_i < w_j \Rightarrow f(p_i) < f(p_j) \tag{5}$$

Thus the function should be decreasing and we can select:

$$f(p) = -\, log(p) \tag{6}$$

The same result is presented in [7] as the ideal weighting function given that one knows the underlying error event probability.

4.2 Implementation comments

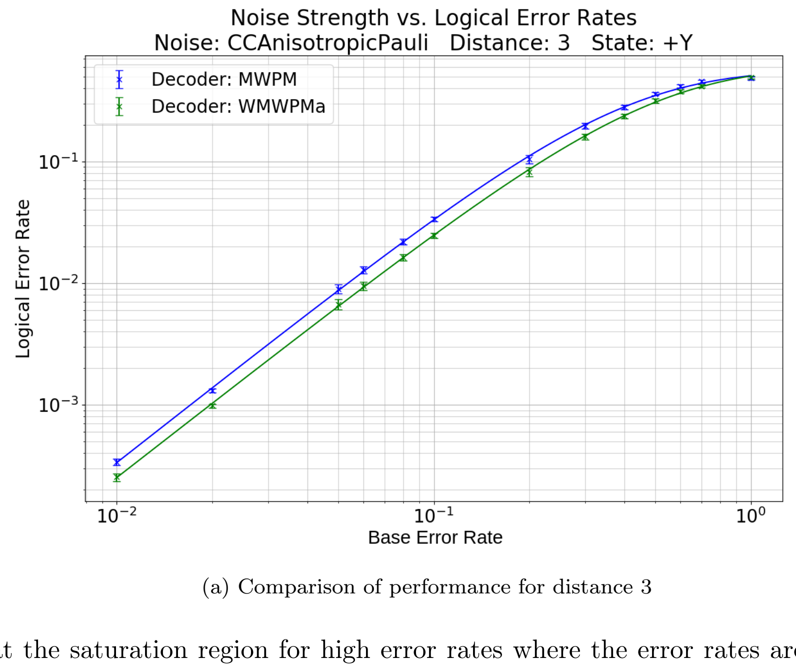
Due to the nature of our code, we need to specify the initial state. Logical state $\pm X_L$ suffers only from logical Z errors and logical state $\pm Z_L$ suffers only from logical X errors. As a result, we shall be selecting the logical $+Y_L$ state ($-Y_L$ would also work) that will be afflicted by both logical X and Z errors. The only way for this state to remain intact is in the presence of a logical error is when a logical X and logical Z error happens simultaneously, which should be only common in high error rates. Note that this is the case only in noise models that are some variant of the depolarising channel and X, Z and Y single-qubit errors are equiprobable.

Additionally, since we want to benchmark the decoders, it is useful to be benchmarking them against the same syndrome. This way our results will be much more reliable when it comes to comparing the performance of the decoders, despite the absolute value of each separately being not precise enough. Even when the error-bars of two logical error rates are not separated, we can still draw comparative conclusions about them. If the syndromes were randomly generated for each decoder though, we should require this error bar separation to reach conclusions.

4.3 Spatially dependent noise model

Firstly, we start with an anisotropic version of the Pauli noise model. Anisotropic in the sense that the error rate will not be the same for every qubit of the lattice.

This can be modelled by establishing some base error rate e and for each error-prone qubit generating a random number r_i drawn from a continuous uniform distribution between 0 and 1. The product er_i will be the probability of this qubit undergoing an error. Thus, the connection between two points in the syndrome lattice will be weighed as $-log(er_i)$ where er_i is the probability of an error happening at the data qubit lying between the two auxiliary qubits. In Figure 6 the results can be seen for distance 3 and 5 using standard MWPM and the customized version of it. The range is selected based on simulation constraints as the lower the base error rate, the more computationally expensive the collection of data becomes. Note though that this range is quite high for a single qubit error rate even in the NISQ era.[2]



(a) Comparison of performance for distance 3

We note that at the saturation region for high error rates where the error rates are for both decoders too high to decode, the state is completely randomized and an logical error rate of 50% is observed for both distance 5 and 3.

For lower error rates we can see that the the specialized decoder (WMWPMa) outperforms standard MWPM by a constant margin. However, because the y axis is a logarithmic axis, this corresponds to a constant ratio. For the $d = 3$ this ratio corresponded to $\sim 35\%$ more logical errors when not using the specialized decoder while for $d = 5$ this ratio was $\sim 65\%$. For an isolated base error rate of 5%, this ratio increased to $\sim 95\%$ for $d = 7$.

Although we cannot draw generalised conclusions, we can see that there is indeed a higher performance as long as we are below the error saturation region up until the base error rate of 0.01. This relative increase in performance seems to be becoming even more significant as the distance of the code d grows. The latter happens because a logical error requires more errors to be misidentified for higher distances d. Since there is a

(b) Comparison of performance for distance 5

Figure 6: Decoding performance of standard MWPM and specialised WMWPMa decoder for Anisotropic Pauli noise in the Code Capacity setting. The error bars represent a 99% confidence interval.

bias towards identifying correctly the errors when using WMWPMa, the larger the distance, the more significant is the effect when compared to MWPM.

Our result is for a lattice that every qubit's error probability may vary up to around one order of magnitude due to the nature of our random selection, which might not be true for a realistic experimental setup. Furthermore, let us recall that the results are in the code capacity setting. Yet, our result due to its simplicity in terms of implementation still stands. Assessing single qubit error rates throughout the lattice and modifying the weights of the MWPM algorithm accordingly is a very efficient way to ensure increase in performance, especially for architectures that may have qubit errors rates with great variance and large code distance.

4.4 Spatially correlated noise model

Secondly, we explore again a variant of the spatially homogeneous Pauli noise model but with nearest neighbour interactions. In particular, the data qubits are subjected to Pauli Noise with probability p and whenever there is an error on some qubit, with some added $(p+p')$ probability there is the same error on a diagonal data qubit neighbour of the qubit.

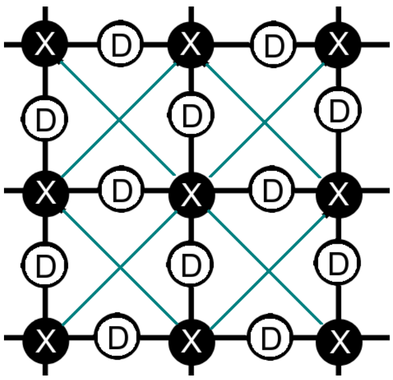


Figure 7: The X syndrome lattice with the data qubits also visible. Every horizontal and vertical edge is assigned a weight of $w_{H,V} = -log(p)$. The coloured diagonal edges are the the additional edges weighted as $w_D = -log(p(p+p'))$

To customize decoding for this noise mode, we are going to add new diagonal edges to our syndrome lattice, which will act as shortcuts when finding the shortest path between nodes. All horizontal and vertical edges are going to be equally weighted as $w_{H,V} = -log(p)$ while these diagonal connections are going to be weighted as $w_D = -log(p(p + p'))$ since the probability of such a correlated event happening is $p_D = p(p + p')$.

We test this noise model using two values of added probability $p' = 0.2, 0.5$. The distance is selected to be 5 such that it is easy to extract data but simultaneously the correlated events have space to occur. Acting always on the same syndrome, we test serially the standard MWPM and the specialized decoder with the added edges according to Figure 7. We present the results in Figure 8.

It is apparent that again the customized decoder outperforms the standard decoder. In fact, the stronger the correlated noise becomes in comparison to the single qubit noise, the more effective it is. This can be verified by comparing the performance "split" for $p' = 0.2$ and $p' = 0.5$ which is greater for the latter.

A question that arises though has to do with whether the assignment of the diagonal weights is indeed ideal. To cross reference this, we select a particular value of error rate p and added correlated probability error p'. For these values, we are going to sweep the decoding parameter which is in fact the ratio of the diagonal weight to the horizontal-vertical weight. The result is presented in Figure 9.

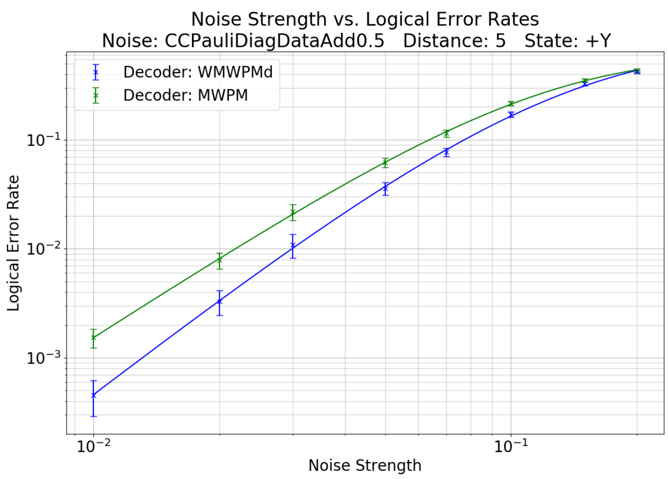
Given that all the parametric decoder and the MWPM algorithm acted on the same syndrome for each trial, the result is rather conclusive despite the error-bars: The theoretically ideal value is indeed the ideal decoding parameter for our decoder.

Also worth noting is that what is important for this decoding parameter is that the logical error rate is not a continuous function but is rather constant in intervals. That is because the value of the decoding parameter indicates how strongly a diagonal path will be prioritized over the indirect horizontal and vertical path. For example, we can see that dp is constant at approximately $[0.75 = \frac{3}{4}, 1 = \frac{4}{4}]$. This means that in the surface code of distance 5, for every decoding parameter in this interval, 3 horizontal/vertical connections will be preferred over 4 diagonal connections but 4 horizontal/vertical connections will not be preferred over 4 diagonal connections. This comes as a result of:

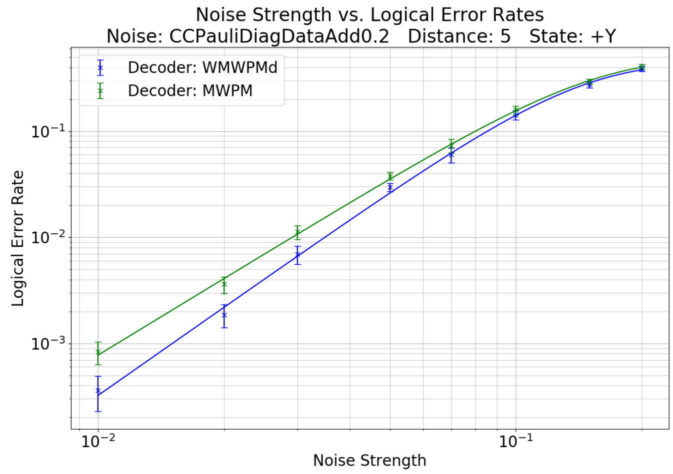
$$\frac{3}{4} < \underbrace{\frac{w_D}{w_{H,V}}}_{\text{decoding parameter}} < \frac{4}{4} \Rightarrow 3\, w_{H,V} < 4\, w_D < 4\, w_{H,V} \tag{7}$$

Since the number of different syndrome connections are finite on a lattice of distance 5, it is understandable why the plot is of this form. We expect that the larger the distance of the code becomes, the more levels of performance this decoding parameter will be able to achieve due to additional combinations of connections. Thus, the less discretized the plot will be and, in order to achieve optimal decoding, the more precisely the ideal decoding parameter will need to be identified.

We can see however that any decoding parameter from 0.75 to 1.5 would work almost equally well. The differences in the performance have to do with how common is a "decoding dilemma", like the one described above, in this specific architecture. A safe assumption would be that any value for the decoding parameter from 0.75 to 1.5 would lead to a probabilistically superior resolution of the most common dilemmas. Picking a value close to the ideal contributes only marginally by resolving some additional uncommon dilemma(s). Of great interest would be how this scales for larger distance codes.



(a) Comparison of performance for added probability $p' = 0.5$



(b) Comparison of performance for added probability $p' = 0.2$ 5

Figure 8: Decoding performance of standard MWPM and specialised WMWPMd decoder for diagonal data qubit correlated noise in the Code Capacity setting. The error bars represent a 99% confidence interval.

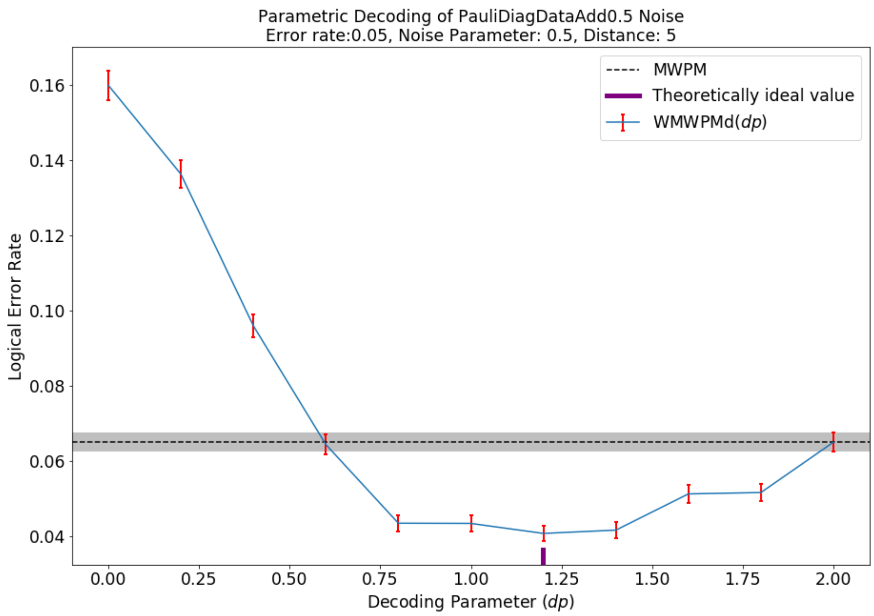


Figure 9: Parametric decoding of the diagonal Noise. The decoding parameter is the ratio $w_D/w_{H,V}$. The shaded region is the error bar of the MWPM. All error bars represent 99% interval of confidence.

5 Conclusion

The surface code performance can benefit in terms of decoding performance by using appropriate weights for the Minimum Weight Perfect Matching algorithm, in cases that the noise has spatial dependence or correlations. We demonstrated that for a case of spatially inhomogeneous Pauli noise the performance enhancement ratio grew with the lattice size whenever we made use of customized decoders by tweaking the weights appropriately. Moreover, we showed that by adding appropriately weighted edges to the syndrome lattice during decoding we were able to drastically decrease the logical error rates when a nearest-neighbour data-qubit noise model was considered. We verified that these diagonal connections were ideally weighted.

As a future outlook, the generalisation of the conclusions drawn here could be further investigated by using more customized software to run error correcting codes on larger lattices. However, of greater interest might be to drop the code capacity setting and attempt to use the techniques employed here to improve fault-tolerance on surface coding via customized decoding. The latter is also our vision and this is why we chose to sacrifice computational power by working on software designed to benchmark fault-tolerance on the surface code.

Acknowledgements

I would like to deeply thank my supervisor Ng Hui Khoon for giving me this opportunity to work on this project and providing guiding feedback throughout the project. I would like to also thank Chai Jing Hao and Mark Myers II for being there daily to answer my questions and help me resolve technical issues. Moreover, many thanks to Dikshant Dulal for proofreading the report. Lastly, I would like to thank my friends and fellow students Smit Chaudhary and Ignacio Fernández Graña for encouraging me to undertake the Honors Program and helping me look for the project.

References

[1] Scott Aaronson and Daniel Gottesman. "Improved simulation of stabilizer circuits". In: *Physical Review A* 70.5 (Nov. 2004). ISSN: 1094-1622. DOI: 10.1103/physreva.70.052328. URL: http://dx.doi.org/10.1103/PhysRevA.70.052328.

[2] K. R. Brown et al. "Single-qubit-gate error below $\mathbf{10^{-4}}$ in a trapped ion". In: *Phys. Rev. A* 84 (3 Sept. 2011), p. 030303. DOI: 10.1103/PhysRevA.84.030303. URL: https://link.aps.org/doi/10.1103/PhysRevA.84.030303.

[3] Jack Edmonds. "Paths, Trees, and Flowers". In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467. DOI: 10.4153/CJM-1965-045-4.

[4] Austin G. Fowler et al. "Surface codes: Towards practical large-scale quantum computation". In: *Physical Review A* 86.3 (Sept. 2012). ISSN: 1094-1622. DOI: 10.1103/physreva.86.032324. URL: http://dx.doi.org/10.1103/PhysRevA.86.032324.

[5] D Gottesman. "The Heisenberg representation of quantum computers". In: (June 1998). URL: https://www.osti.gov/biblio/319738.

[6] Chai Jing Hao. "Assessing Fault-Tolerance Conditions for Surface Code Implemented With Noisy Devices". PhD thesis. Centre for Quantum Technologies, National University of Singapore, 2020.

[7] Naomi H. Nickerson and Benjamin J. Brown. "Analysing correlated noise on the surface code using adaptive decoding algorithms". In: *Quantum* 3 (Apr. 2019), p. 131. ISSN: 2521-327X. DOI: 10.22331/q-2019-04-08-131. URL: http://dx.doi.org/10.22331/q-2019-04-08-131.

[8] Joschka Roffe. "Quantum error correction: an introductory guide". In: *Contemporary Physics* 60.3 (July 2019), pp. 226–245. ISSN: 1366-5812. DOI: 10.1080/00107514.2019.1667078. URL: http://dx.doi.org/10.1080/00107514.2019.1667078.