
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Üstün, Gözde; Morello, Andrea; Devitt, Simon

Single-step parity check gate set for quantum error correction

Published in:
Quantum Science and Technology

DOI:
[10.1088/2058-9565/ad473c](https://doi.org/10.1088/2058-9565/ad473c)

Published: 01/07/2024

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:
Üstün, G., Morello, A., & Devitt, S. (2024). Single-step parity check gate set for quantum error correction. *Quantum Science and Technology*, 9(3), 1-27. Article 035037. <https://doi.org/10.1088/2058-9565/ad473c>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

PAPER • OPEN ACCESS

Single-step parity check gate set for quantum error correction

To cite this article: Gözde Üstün *et al* 2024 *Quantum Sci. Technol.* **9** 035037

View the [article online](#) for updates and enhancements.

You may also like

- [Density matrix simulation of quantum error correction codes for near-term quantum devices](#)
Chungheon Baek, Tomohiro Ostuka, Seigo Tarucha et al.
- [Magnetic field and microstructural effects on the electrical capacitance and resistance of a quadrupolar electrical capacitor based on cotton fabrics and carbonyl iron microparticles](#)
Ioan Bica and Eugen Mircea Anitas
- [Quantum error correction for beginners](#)
Simon J Devitt, William J Munro and Kae Nemoto

Quantum Science and Technology



PAPER

Single-step parity check gate set for quantum error correction

OPEN ACCESS

RECEIVED
26 June 2023

REVISED
26 April 2024

ACCEPTED FOR PUBLICATION
3 May 2024

PUBLISHED
21 May 2024

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Gözde Üstün^{1,*} , Andrea Morello¹  and Simon Devitt³ 

¹ Centre for Quantum Computation and Communication Technology, School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, New South Wales 2052, Australia

² Centre for Quantum Software and Information, School of Computer Science, Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Sydney, Australia

³ InstituteQ, Aalto University, Espoo 02150, Finland

* Author to whom any correspondence should be addressed.

E-mail: g.ustun@unsw.edu.au

Keywords: quantum error correction, fault-tolerant quantum computation, quantum control

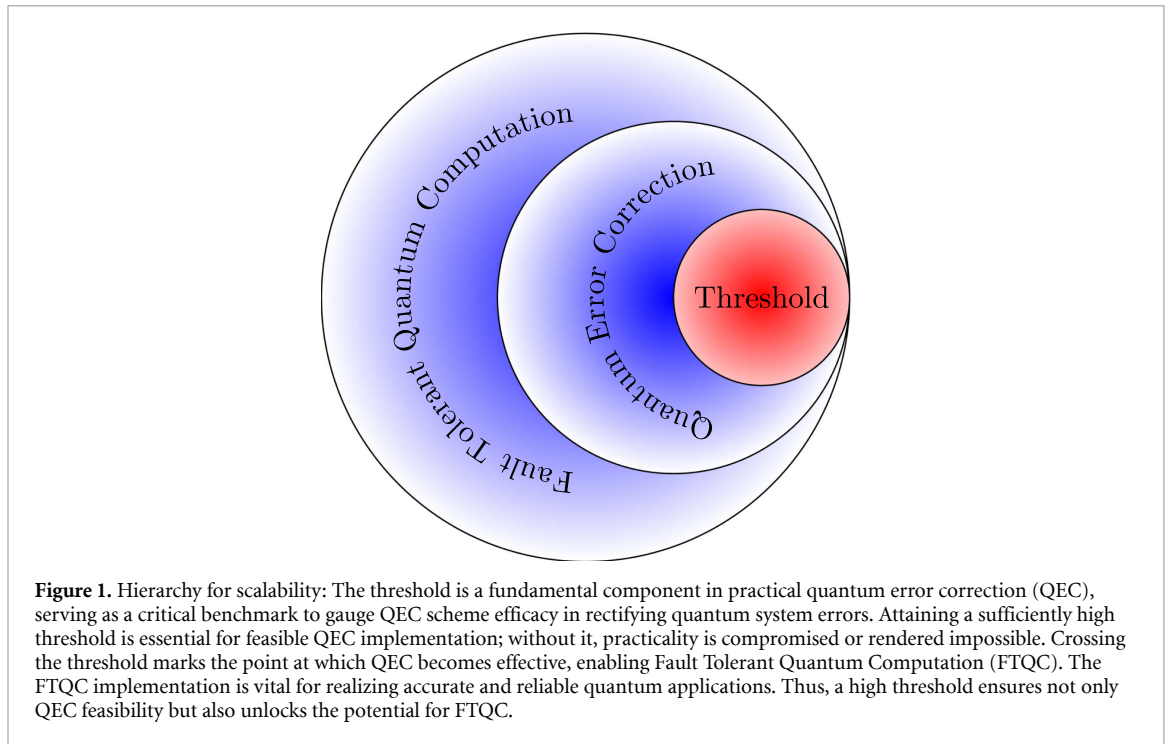
Abstract

A key requirement for an effective quantum error correction (QEC) scheme is that the physical qubits have error rates below a certain threshold. The value of this threshold depends on the details of the specific QEC scheme, and its hardware-level implementation. This is especially important with parity-check circuits, which are the fundamental building blocks of QEC codes. The standard way of constructing the parity check circuit is using a universal set of gates, namely sequential CNOT gates, single-qubit rotations and measurements. We exploit the insight that a QEC code does not require universal logic gates, but can be simplified to perform the sole task of error detection and correction. By building gates that are fundamental to QEC, we can boost the threshold and ease the experimental demands on the physical hardware. We present a rigorous formalism for constructing and verifying the error behavior of these gates, linking the physical measurement of a process matrix to the abstract error models commonly used in QEC analysis. This allows experimentalists to directly map the gates used in their systems to thresholds derived for a broad-class of QEC codes. We give an example of these new constructions using the model system of two nuclear spins, coupled to an electron spin, showing the potential benefits of redesigning fundamental gate sets using QEC primitives, rather than traditional gate sets reliant on simple single and two-qubit gates.

1. Introduction

Being able to scale quantum computers is a necessary condition for useful quantum algorithms [1]. Algorithms with proven speed up, such as Shor's factorization algorithm [2], quantum simulation [3] or the quantum algorithm for solving linear systems of equations [4] only become useful when implemented on thousands of near-perfect qubits. As it is unlikely that physical error rates will be vanishingly small in the near term, the addition of quantum error correction (QEC) and fault-tolerance is required. For practical QEC, the critical metric is the fault-tolerance threshold. This is the physical error rate below which QEC becomes effective and corrects more errors than it introduces (figure 1). Therefore the threshold value, determined by the QEC scheme's structure and subsequent fault-tolerant circuit and hardware implementations [5], is the highest acceptable physical error per gate for each qubit.

The surface code, a well-performing and experimentally feasible code [6–9], exhibits a high threshold, dependent on the assumed noise model. A noise model incorporates assumptions about error channels like qubit loss and leakage, affecting the fault-tolerance threshold. Making assumptions about how error syndromes are measured in the system, commonly referred to as code capacity noise model, phenomenological noise model, and circuit-based noise model [10], can lead to significantly different fault-tolerant threshold values. For the surface code, which involves extracting syndromes based on the parity of four-body operators, different noise models consider different methods for measuring the parity of these operators.



The code capacity and phenomenological noise models rely on natural four-body measurements, while the circuit-based model constructs four-body parity check measurements using one-body Pauli measurements and standard quantum gates. This approach yields the lowest code threshold due to physical error accumulation on standard quantum gates throughout the circuit. The circuit-based noise model is the most experimentally precise analysis of the surface code, as most experimental qubit systems lack direct access to four-body measurements.

In this study, rather than enhancing thresholds with current sources specifically designed for universal computation, we redesign the universal gates to be inherently suited for QEC implementation. Focusing on gates fundamental to QEC, as opposed to universal computation, enhances the threshold and simplifies experimental scalability challenges. This modification is deemed necessary as any large-scale quantum computer inherently functions as a QEC machine.

For error-correcting quantum computing, occasional use of traditionally universal gates suffices as primitives. However, QEC necessitates defining a new gate set, achieved by modifying the existing universal gate set. Then the challenge lies in how an experimental system can validate implementability of the new gate set in the lab, ensuring its effective use to raise the QEC threshold.

In this work, we present a rigorous formalism for constructing and verifying a native gate set, which we denote as the single-step parity check (SSPC) gate set, designed specifically for QEC circuits. We apply this technique to the two-body parity check circuits necessary for implementing the Honeycomb code [11]. We present the SSPC gates in the context of nuclear spin qubits in silicon [12], since they offer a native, direct two-body measurements via a bound electron spin [13]. However, this formalism applies more broadly and could be used for hardware systems that can naturally implement high dimensional parity check gate primitives.

The organization of this paper is as follows: section 2.1 briefly reviews the concept of multi-qubit measurements and introduce notations that are used for the later part of the paper. In section 2.2, the Honeycomb code is explained. In section 3 the drawback of the circuit-based noise model is shown numerically by analyzing the experimental data [13]. In section 4, we introduce the SSPC gate set and in section 4.1, we theoretically compare two-body parity check circuits constructed both using the universal gate set and the SSPC gate set in the presence of noise. In section 4.2, we provide an example system of silicon spin qubits to implement the SSPC gate set. In section 5, we summarise the paper.

2. Preliminaries

2.1. The concept of making multi qubit parity check measurement

Even in a single qubit, one can define the 1 state as odd parity, and the 0 state as even parity, i.e. a single qubit measurement already constitutes a parity check. In this case, it will be labelled M_{pp1} , where pp refers to Pauli

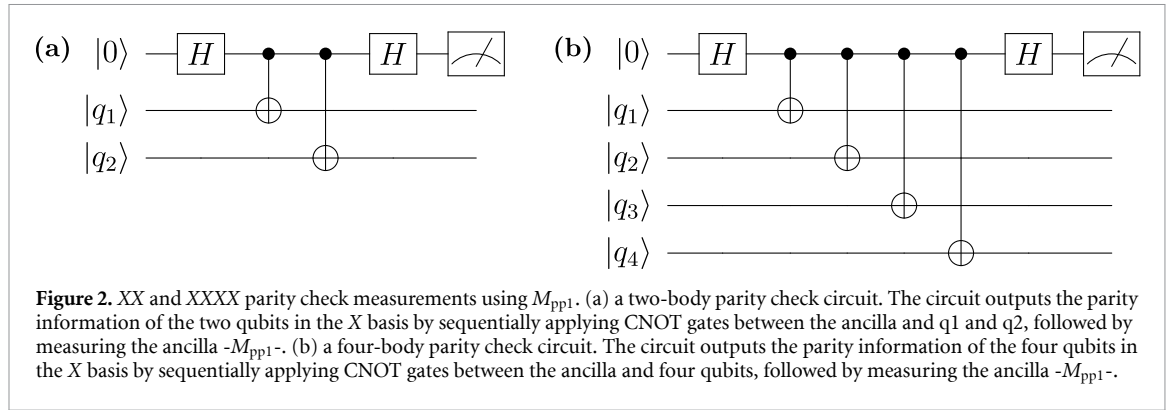


Figure 2. *XX* and *XXXX* parity check measurements using M_{pp1} . (a) a two-body parity check circuit. The circuit outputs the parity information of the two qubits in the *X* basis by sequentially applying CNOT gates between the ancilla and q1 and q2, followed by measuring the ancilla $-M_{pp1}$ -. (b) a four-body parity check circuit. The circuit outputs the parity information of the four qubits in the *X* basis by sequentially applying CNOT gates between the ancilla and four qubits, followed by measuring the ancilla $-M_{pp1}$ -.

Product and M_{pp1} refers to measuring in one Pauli basis, for example measuring in the *Z* basis. This can be generalized beyond a single qubit and becomes particularly useful in QEC. The idea of parity checks in QEC is to have just parity information of multiple data qubits over one (ancilla) qubit without revealing any other information. In other words, we are constructing $M_{ppi}, i = 2, 3, 4, \dots$ from M_{pp1} . To construct M_{ppi} from M_{pp1} for a valid quantum observable U , we apply the controlled- U gates between the ancilla qubit and the data qubits sequentially. For example, for a parity check measurement in the *X* basis, the controlled- U gates are simply CNOT gates.

While any unitary can be used for the parity check, we provide an example using well-known parity check measurements in QEC. For instance, with an *XX* parity check measurement ($XX = X \otimes X$), we use two CNOT gates; for an *XXXX* parity check measurement ($XXXX = X \otimes X \otimes X \otimes X$), we use four CNOT gates. To measure the system over one (ancilla) qubit, we use M_{pp1} ⁴. Let us look at figure 2(a) closer to understand what a parity check does to a state.

In figure 2(a), we started with an initial state of the ancilla qubit which is $|0\rangle$. The initial combined state of the system is $|0\rangle \otimes |\psi_{in}\rangle$ in which $|\psi_{in}\rangle$ is a general state over $|q_1\rangle$ and $|q_2\rangle$. Then, applying the Hadamard gate transforms the initial combined state of the system, $|0\rangle \otimes |\psi_{in}\rangle$, as follows:

$$|0\rangle|\psi_{in}\rangle \xrightarrow{\text{Hadamard}} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |\psi_{in}\rangle. \tag{1}$$

After the Hadamard gate, once we apply the CNOT gates sequentially, we have the transformation of (1) to (2):

$$\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) |\psi_{in}\rangle \xrightarrow{\text{CNOT's}} \frac{1}{\sqrt{2}} (|0\rangle|\psi_{in}\rangle + |1\rangle XX|\psi_{in}\rangle). \tag{2}$$

We apply the last Hadamard gate and have the transformation of (2) as follows:

$$\frac{1}{\sqrt{2}} (|0\rangle|\psi_{in}\rangle + |1\rangle XX|\psi_{in}\rangle) \xrightarrow{\text{Hadamard}} \frac{1}{2} |0\rangle (|\psi_{in}\rangle + XX|\psi_{in}\rangle) + \frac{1}{2} |1\rangle (|\psi_{in}\rangle - XX|\psi_{in}\rangle). \tag{3}$$

The final step of the circuit is to measure the ancilla qubit. After this step, if:

$$\begin{aligned} \text{ancilla} = |0\rangle &\Rightarrow \frac{|\psi_{in}\rangle + XX|\psi_{in}\rangle}{2} \cong |\psi_{out}\rangle \\ \text{ancilla} = |1\rangle &\Rightarrow \frac{|\psi_{in}\rangle - XX|\psi_{in}\rangle}{2} \cong |\psi_{out}\rangle \end{aligned} \tag{4}$$

where \cong means up to a renormalization factor. The symmetries of $|\psi_{out}\rangle$ are If:

$$\begin{aligned} \text{measured}|0\rangle &\Rightarrow XX|\psi_{out}\rangle = XX \left(\frac{|\psi_{in}\rangle + XX|\psi_{in}\rangle}{2} \right) \\ &= \frac{XX|\psi_{in}\rangle + |\psi_{in}\rangle}{2} \\ &= |\psi_{out}\rangle \end{aligned} \tag{5}$$

⁴ Any valid quantum observable can be used for the parity check, determined by the operator on the target side of the controlled gate.

If:

$$\begin{aligned}
 \text{measured}|1\rangle \Rightarrow XX|\psi_{\text{out}}\rangle &= XX \left(\frac{|\psi_{\text{in}}\rangle - XX|\psi_{\text{in}}\rangle}{2} \right) \\
 &= \frac{XX|\psi_{\text{in}}\rangle - |\psi_{\text{in}}\rangle}{2} \\
 &= - \left(\frac{|\psi_{\text{in}}\rangle - XX|\psi_{\text{in}}\rangle}{2} \right) \\
 &= -|\psi_{\text{out}}\rangle
 \end{aligned} \tag{6}$$

Hence, dependent on the measurement result, we have projected our state into eigenstates of the XX operator with eigenvalues $+1$ or -1 , with the ancilla measurement result dictating which eigenstate we have projected into. If the input was already in an eigenstate with eigenvalues $+1$ or -1 , we will simply obtain a measurement result on the ancilla that indicates the parity of the input state, without changing the state itself. We therefore have the parity information of the multi qubit system by using CNOT gates and a one-body measurement, (CNOT's + M_{pp1}).

Multi-body measurements are used in different aspects of quantum information/computation. In QEC, stabilizer measurements are the important examples of multi-body measurements, that occur over the operators that define the stabilizers of the QEC code. However, for example, instead of making direct XX parity check measurement, called M_{pp2} , we perform this measurement by using CNOT's + M_{pp1} and single qubit rotations. Here M_{pp2} refers to the fact that the operator we are measuring, XX , is a Pauli operator with weight two. In semiconductor spin qubits, single-triplet readout is also a multi-body measurement over 2 qubits. However, we are not using it to measure the parity of a multi-qubit Pauli operator, so we do not classify it as M_{pp2} .

Measuring a qubit is, in fact, a one-body parity measurement, allowing the unique definition of states. However, when measuring multiple qubits, our goal is to constrain the global space into subspaces where the symmetry of the measured operator is enforced. Consider the example in figure 2(a), where the input state is chosen as:

$$|\psi_{\text{in}}\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \tag{7}$$

where α, β, γ and δ are arbitrary complex amplitudes that define a unique state in this four dimensional Hilbert space. If we restrict the space by enforcing a symmetry, i.e. by running the circuits in figure 2(a), prior to measurement of the ancilla, the state of the three qubits before measurement, $|\psi_{\text{bm}}\rangle$, is:

$$|\psi_{\text{bm}}\rangle = |0\rangle \left(\frac{1}{2} (I + XX) |\psi_{\text{in}}\rangle \right) + |1\rangle \left(\frac{1}{2} (I - XX) |\psi_{\text{in}}\rangle \right). \tag{8}$$

Here $|0\rangle$ and $|1\rangle$ are the states of the ancilla qubit. We know that by measuring the ancilla state will project the state into an eigenstate of XX . If we assume that we measure the ancilla in $|0\rangle$ and project our state into the $+1$ eigenstate of XX , we can write our output state $|\psi_{\text{out}}\rangle$ as

$$|\psi_{\text{out}}\rangle = \frac{1}{2} (\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle) + \frac{1}{2} (\alpha|11\rangle + \beta|10\rangle + \gamma|01\rangle + \delta|00\rangle). \tag{9}$$

By grouping them

$$|\psi_{\text{out}}\rangle = (\alpha + \delta) \left(\frac{|00\rangle + |11\rangle}{2} \right) + (\beta + \gamma) \left(\frac{|01\rangle + |10\rangle}{2} \right) \tag{10}$$

we can rewrite equation (10) as follows:

$$\begin{aligned}
 \frac{(\alpha + \delta)}{\sqrt{2}} &= \alpha' \text{ and } \frac{(|00\rangle + |11\rangle)}{\sqrt{2}} = |\hat{0}\rangle \\
 \frac{(\beta + \gamma)}{\sqrt{2}} &= \beta' \text{ and } \frac{(|01\rangle + |10\rangle)}{\sqrt{2}} = |\hat{1}\rangle
 \end{aligned} \tag{11}$$

As a result, we have:

$$|\psi_{\text{out}}\rangle = \alpha' |\hat{0}\rangle + \beta' |\hat{1}\rangle \tag{12}$$

which is effectively a new qubit. Hence, by enforcing a symmetry on our original two-qubit system, we have reduced it to an effective one qubit system, where our new basis states, $\{|\hat{0}\rangle, |\hat{1}\rangle\}$, are both $+1$ eigenstates of

XX. Making multi-body parity measurements enforces the symmetry of the system and reduces the number of degrees of freedom. A two-body parity measurement over a two qubit system reduces the result into effective one-qubit. This can be generalised to an N qubit system: for every symmetry we enforce, by performing a measurement of an arbitrary operator, we reduce the effective number of qubits in the system by one [5, 14].

2.2. Honeycomb code

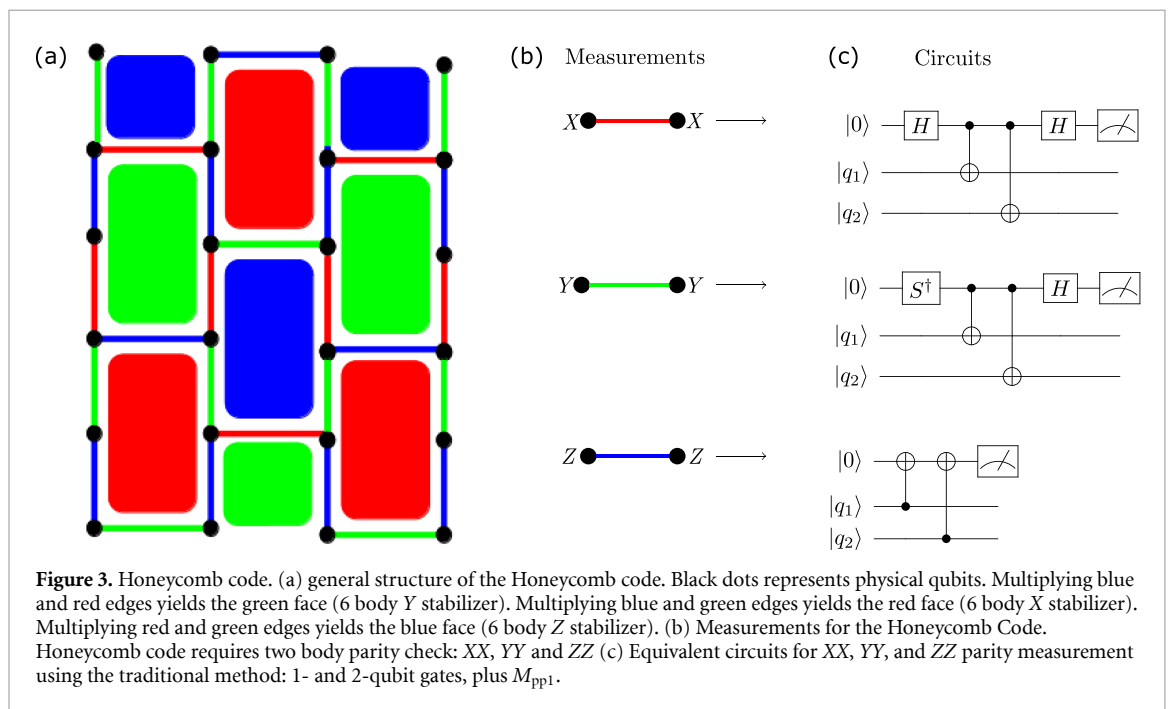
In 2020, the Honeycomb code [15], a variant of Floquet codes [16], was introduced as a novel QEC code⁵. The Honeycomb code stands as one of the more recent advancements in topological code constructions. It involves qubits positioned on a 2D lattice, forming a fixed hexagonal configuration. The distinguishing feature of the Honeycomb code, in contrast to the surface code, is its reliance on two-body parity checks, specifically XX , YY , and ZZ measurements, instead of requiring $XXXX$ and $ZZZZ$ measurements. Figure 3(c) provides the equivalent circuit for performing XX and ZZ parity check measurements in the Honeycomb code.

It was also shown that this code exhibited comparable thresholds to the surface code [18], in contrast to Gauge codes which can be realised using two-body parity checks, but whose stabilisers are generally much higher weights [19]. Since the Honeycomb code requires only two-body parity measurements, M_{pp2} , which may be achievable experimentally in some systems naturally.

In [11], Gidney and Fowler demonstrated that when considering a scenario where a natural M_{pp2} measurement is available from the hardware, the error modeling mirrors a phenomenological noise model. This involves applying a perfect multi-qubit measurement followed by error mapping according to the equation:

$$\{I, X, Y, Z\}^{\otimes 2} \times \{\text{flip, no flip}\} \tag{13}$$

with probability, p . This error map allows us to differentiate between two types of errors: quantum errors and classical errors. The first set of brackets represents the quantum errors and it allows us to have one Pauli error on one of the data qubits. The second set of brackets represents the classical error and it allows us to have one classical bit flip error which is the bit flip error on the classical measurement result on the ancilla qubit. With this map, the threshold of the Honeycomb code was found to be approximately 2%. This threshold is commensurate with the phenomenological threshold for the surface code, but unlike the surface code, some hardware systems may have the possibility of realising M_{pp2} operations, while it is unlikely that a natural M_{pp4} measurement will be found in any current hardware systems.



⁵ This code was originally introduced by Kitaev in 2006 [17], but its use as a QEC code was demonstrated in [15].

This means that if we can have a natural two-body measurement that satisfies the error model of Gidney and Fowler, we will have a three times higher threshold for the Honeycomb code while maintaining a 2D nearest-neighbour hardware architecture. Furthermore, the latest study by Gidney [20], shows that if the surface code is run by using natural two-body measurements, the threshold of the surface code will be similarly increased.

The next question is: how can we assess whether a physical system allows a natural implementation of natural M_{pp2} operations? For this purpose, we outline a workflow to map an experimental process matrix to a QEC-specific error model for checking if a real system's error behavior matches the assumptions of a threshold analysis. We develop and validate a native gate set tailored for QEC, focusing on higher dimensional unitaries required for syndrome extraction circuits, termed the SSPC gate set. Applying this technique to two-qubit parity check circuits for the Honeycomb code implementation, we illustrate the SSPC gates in the context of spin qubit systems [12, 13].

3. Analyzing the XX parity check circuit from experimental data in the context of QEC

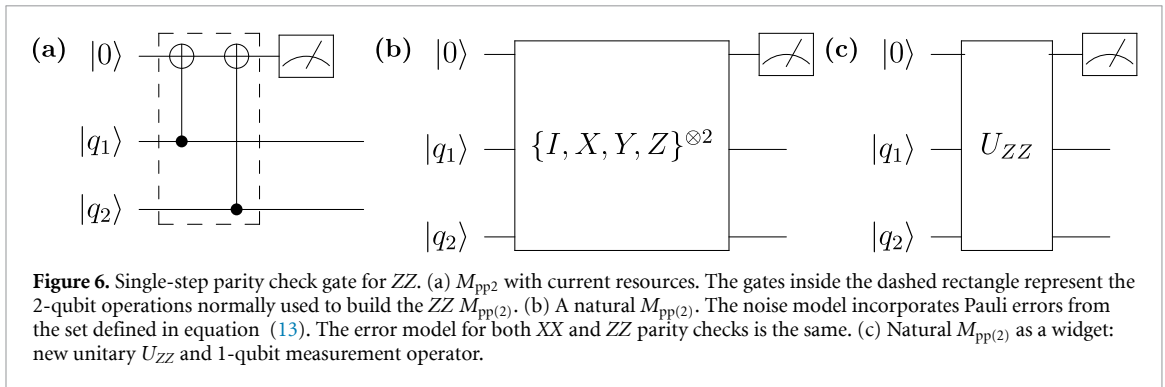
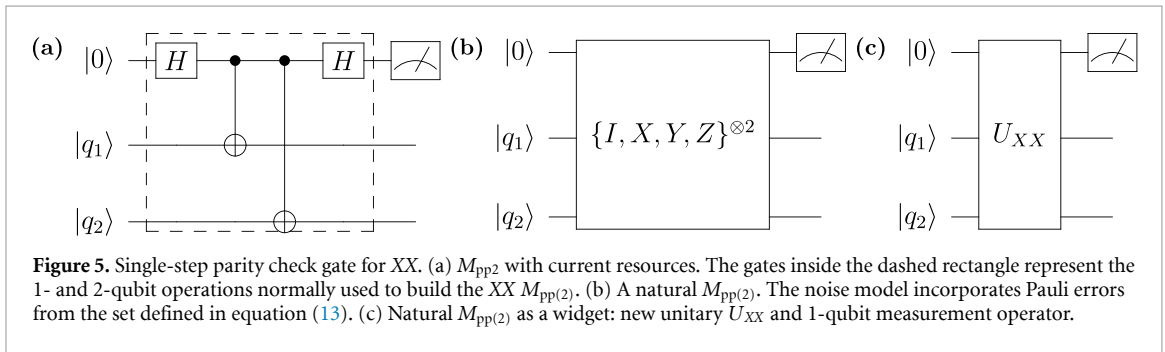
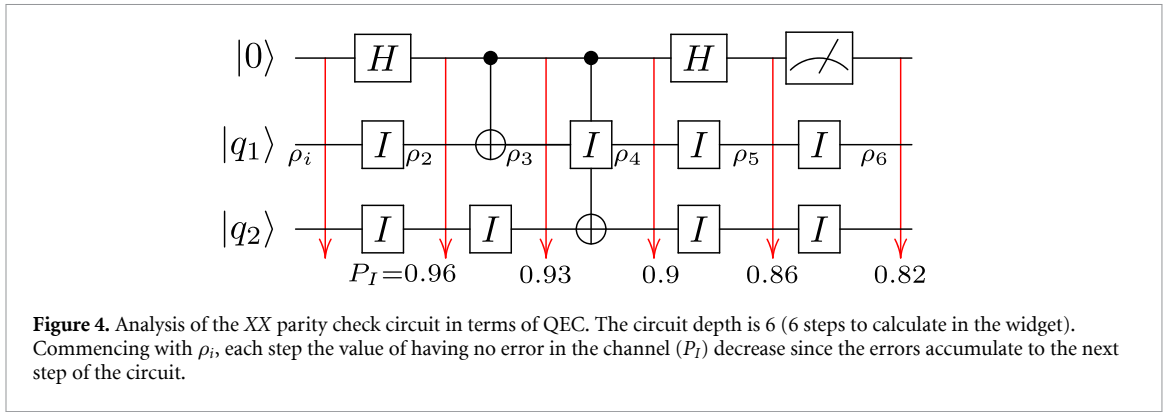
Assessments of hardware systems have primarily focused on single- and two-qubit gate fidelities, assuming fidelity reflects operational perfection. However, fidelity is a numerical representation lacking information on error type, and it does not correspond to gate error rates. In QEC, we analyze errors using stochastic Pauli channels, focusing on error types and values for 1- and 2-qubit gates instead of relying on gate fidelities. Furthermore, relying solely on the error values for these gates is insufficient to determine the total circuit error rate and, consequently, the overall circuit perfection rate due to error propagation in the circuit.

Here we numerically demonstrate (to the best of our knowledge, for the first time) the loss in the circuit by calculating the perfection rate at each step when utilizing one-body measurement (M_{pp1}) with 1- and 2-qubit gates, based on experimental data. The detailed explanation of the perfection rate calculation A.1, including a comprehensive tutorial A.4, is available in the appendices. Our algorithm A.3 allows experimentalists to directly map the gates used in their systems to thresholds derived for a broad-class of QEC codes.

In this section, we use data from [13] to show how the output perfection rate decreases for XX parity check circuit which is the smallest parity check measurement that we can use to implement a topological error correcting code. This study showcases the universal quantum logic operations performed on a silicon nanoelectronic device utilizing a pair of ion-implanted ^{31}P donor nuclei. This device, consisting of two phosphorus nuclei and one electron spin system, is henceforth referred to as the '2P1e' device. Researchers precisely characterized the quantum operations by using gate set tomography (GST) [21, 22]. They achieved one-qubit gates with average gate fidelities up to 99.95(2)%, two-qubit gates with an average gate fidelity of 99.37(11)% and two-qubit preparation/measurement fidelities of 98.95(4)%. These results indicate that nuclear spins in silicon are close to the performance required by fault-tolerant quantum processors. Figure 4 shows the decrease of perfection rate for the XX parity check measurement which is constructed by the circuit based model. Another important point is the identity gates. When we apply a single qubit gate on a multi-qubit circuit, the qubits which have no applied gate should stay stable and this is represented by identity gates. Identity gates are also as important as other single-qubit gates since they contribute to the errors. As each step progresses in figure 4, errors accumulate. Initially, the system is in state ρ_i . Then, a Hadamard gate is applied to the first qubit and an identity is applied to the other two, resulting in state ρ_2 from ρ_i . The final density matrix for ρ_2 is calculated, in terms of having experienced certain errors in a QEC block, resulting in a perfection rate of 0.96.

The next step, ρ_3 , includes errors from not only CNOT and identity operations, but also from ρ_2 . When the process continues, the final perfection rate is found to be 0.82. The perfection rate between the application of the first operation—in this case, the Hadamard gate on the ancilla qubit and the identities on the $|q_1\rangle$ and $|q_2\rangle$ —and the measurement operation decreased by 14%, from 0.96 to 0.82. This striking observation highlights how, even in a state-of-the-art experiment with average gate fidelities above 99%, constructing a two-body parity check circuit out of universal logic gates results in a drastic reduction of the perfection rate at the end of circuit.

With the goal of optimizing the perfection rate for an M_{pp2} operation, we thus aim to design full unitaries directly, instead of composing them from 1- and 2-qubit gates. We construct these new unitaries by modifying the universal gate set. These unitaries are fundamental to QEC rather than universal computation. By using these gates, we can increase the threshold and decrease the experimental challenge to the scalability. We call these gates as single step-parity check gate set. These gates are actually multi-body parity measurement gates. This is described in section 4.



4. Single-Step Parity Check gate set and M_{pp2}

Single-step parity check (SSPC) gates are a version of pulse optimization where we optimize for the full unitary needed for QEC. In this case, we reduce the number of steps and instead of having six different steps, we have single step for the same circuit. From figures 5(a) and 6(a), the unitaries are found to be equations (14) and (15) and they belong to the Clifford group. Apart from the new unitary, we have a new device which can be called a natural M_{pp2} . It is a widget that includes one of those new unitaries, and a measurement operator. It receives 3 qubits as input, using one of them as an ancilla, and returns the parity information by measuring the ancilla qubit. In the next section, we compare the SSPC gate and the decomposed parity check circuit theoretically in the presence of noise.

$$U_{XX} = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{14}$$

$$U_{ZZ} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (15)$$

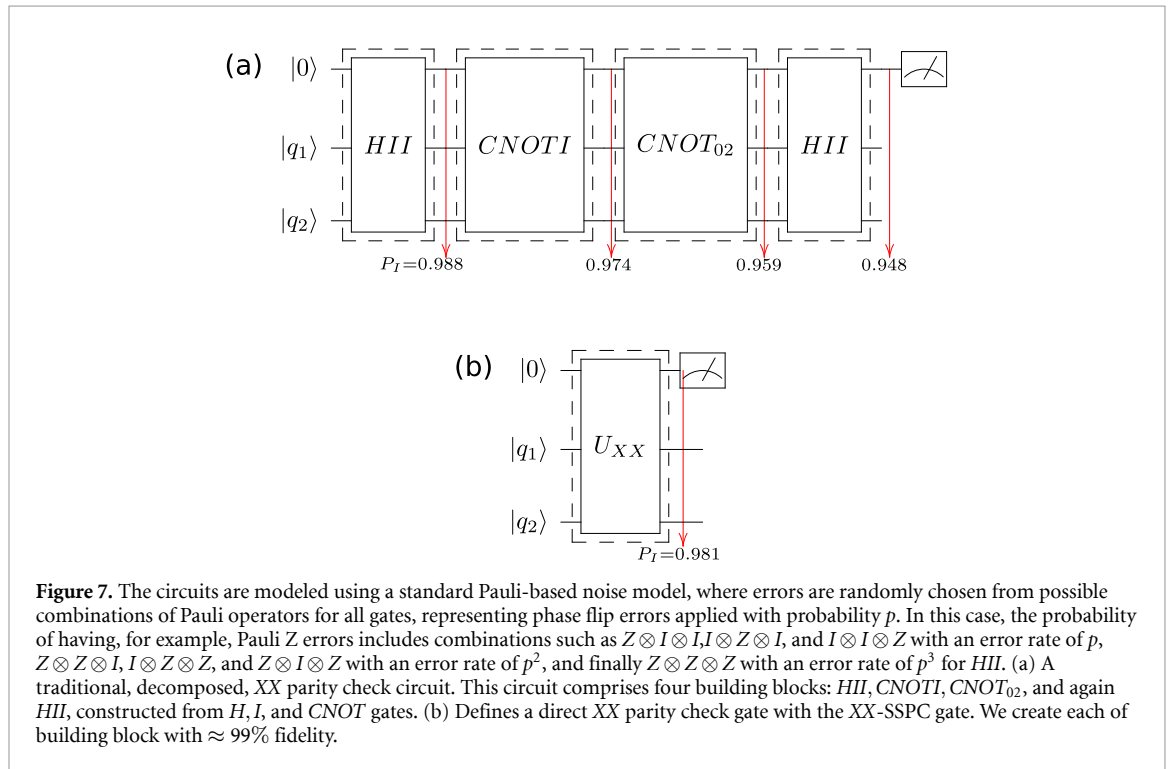
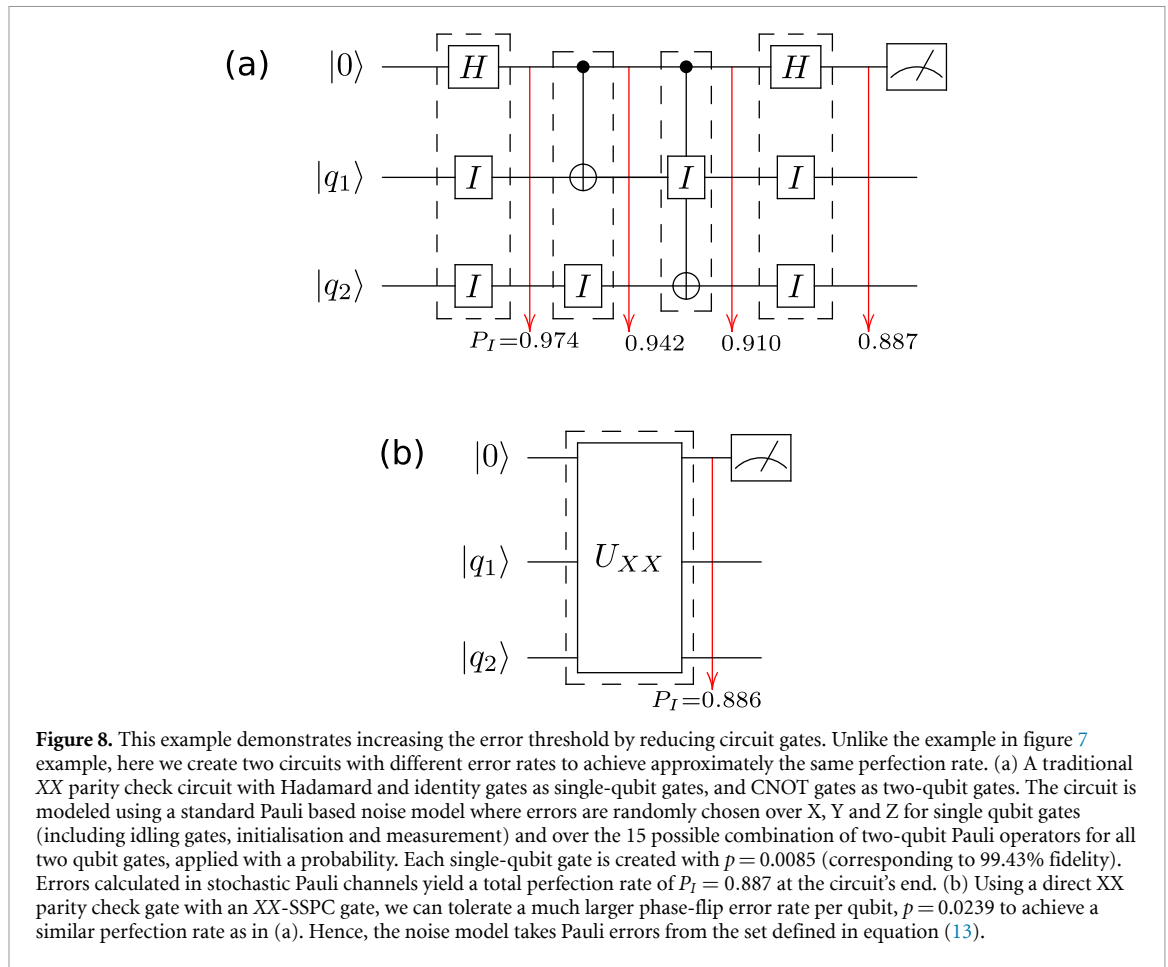


Figure 7. The circuits are modeled using a standard Pauli-based noise model, where errors are randomly chosen from possible combinations of Pauli operators for all gates, representing phase flip errors applied with probability p . In this case, the probability of having, for example, Pauli Z errors includes combinations such as $Z \otimes I \otimes I$, $I \otimes Z \otimes I$, and $I \otimes I \otimes Z$ with an error rate of p , $Z \otimes Z \otimes I$, $I \otimes Z \otimes Z$, and $Z \otimes I \otimes Z$ with an error rate of p^2 , and finally $Z \otimes Z \otimes Z$ with an error rate of p^3 for HII . (a) A traditional, decomposed, XX parity check circuit. This circuit comprises four building blocks: HII , $CNOTI$, $CNOT_{02}$, and again HII , constructed from H , I , and $CNOT$ gates. (b) Defines a direct XX parity check gate with the XX -SSPC gate. We create each of building block with $\approx 99\%$ fidelity.

4.1. Comparison of decomposed XX parity check circuit and XX -SSPC circuit in the presence of noise

In this section, we will show how the perfection rate differs in the presence of noise between decomposed parity check circuit and SSPC circuit. For this purpose, we take the circuits defined in figure 7. In figure 7(a), the circuit has four building blocks constructed from Hadamard, Identity, and CNOT gates: $HII = H \otimes I \otimes I$, $CNOTI = CNOT \otimes I$, $CNOT_{02}$ (CNOT gate between the ancilla and the second data qubit), and $HII = H \otimes I \otimes I$. In figure 7(b) the circuit contains only one building block, the U_{XX} gate. Each noisy operator can be modeled by perfect operator plus some errors to be mapped in the circuit. We simulated process matrices such that each gate is modeled by a noisy channel with a 0.00375 phase flip error rate. Then we calculate the average gate fidelity from these noisy channels and we find that each building block in the circuit will have $\approx 99\%$ fidelity. Note that if we change the error type, the amount of error required to maintain the same fidelity will also change. The reason for selecting this error type is because phase flip errors are the predominant sources of error for most solid-state hardware systems, especially in semiconductor spin qubits such as the one described in [13]. Overall, we have on one side four building block with 99% fidelity, and on the other we have one building block with 99% fidelity. We apply our workflow A.1 to find the perfection rate of each circuit defined in figure 7.

As it is seen from the result, since the traditional, decomposed, parity check circuit requires four steps (four building blocks), the errors in each step accumulate. However, with a SSPC gate, there is no more than one building block so there is no place for errors to accumulate. For this reason, the perfection rate remains high. As a further illustration, in figure 8 we show a simulation where each single qubit gate (Hadamard and Identity gates) has 99.43% fidelity, and the CNOT gate experiences the single qubit errors as a tensor product of the amount $(I \otimes X, I \otimes Y \dots)$. We chose this fidelity value because it is the surface code fault-tolerance threshold [6]. To achieve this fidelity, we used an error rate of $p = 0.0085$ for the phase flip error in each single-qubit gate. As depicted in figure 8, the resulting perfection rate at the circuit's end is 0.887. This perfection rate is obtained by utilizing gates with the specified fault tolerance error value.



To attain an equivalent perfection rate for the XX-SSPC gate, comparable to the decomposed parity check circuit, we can increase the error rate to $p = 0.0239$ for the phase flip error per qubit. Implementing an XX gate with this error rate yields a perfection rate $P_I = 0.886$, as illustrated in figure 8(b). As a result once the parity check circuit is created with SSPC gate at higher error rates than the 1- and 2-qubit gates, the same perfection rate can be maintained. This simple example shows how SSPC gates can drastically boost the error threshold value for QEC.

4.2. Practical implementation of SSPC Gates

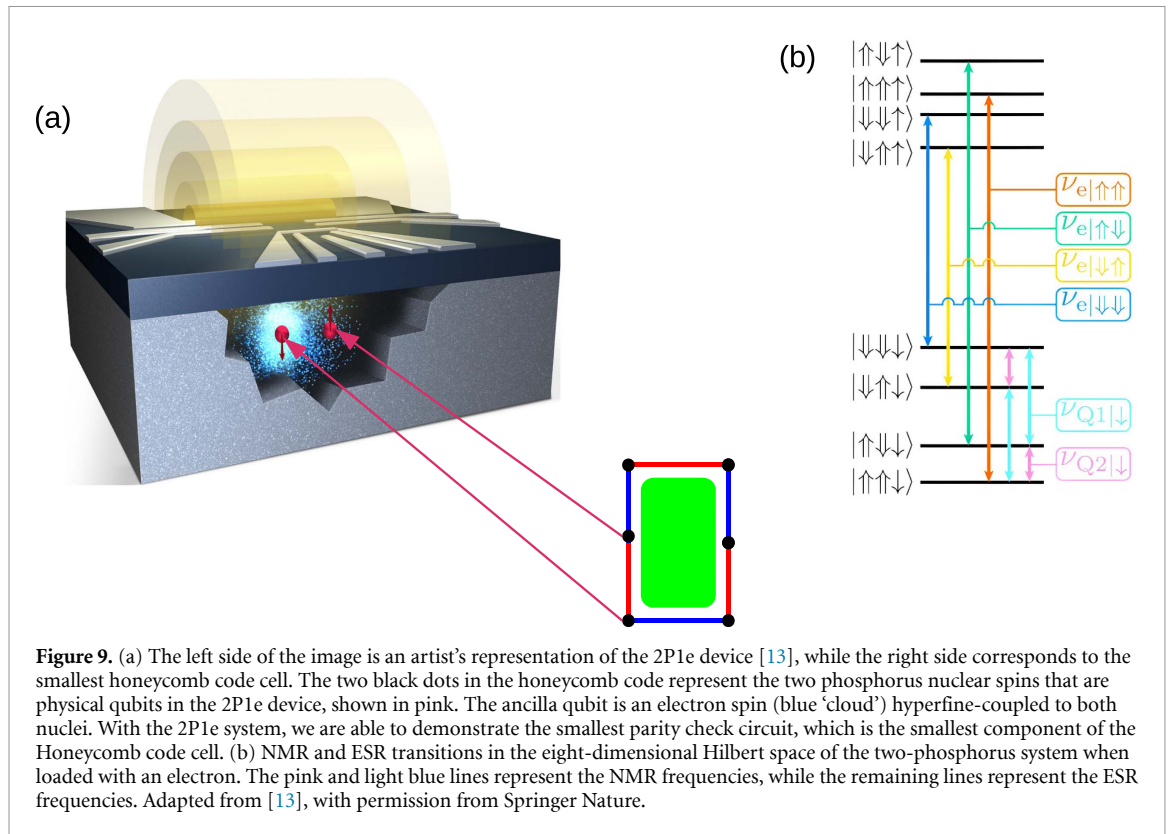
The theoretical discussion in the preceding section illustrates the advantages afforded by implementing parity checks using a single-step M_{pp2} operation instead of breaking it down to a sequence of individual gates. We now illustrate a practical example of how a SSPC gate could be implemented in a real physical system.

Let us consider a three-spin system consisting of an electron spin, acting as the ancilla, and two nuclear spins, acting as the qubits. Their drift Hamiltonian in the laboratory frame takes the general form:

$$\vec{H} = -\gamma_e B_0 \hat{S}_z - \gamma_n B_0 (\hat{I}_{z1} + \hat{I}_{z2}) + A_1 \vec{S} \cdot \vec{I}_1 + A_2 \vec{S} \cdot \vec{I}_2, \tag{16}$$

where $\vec{S} = [\hat{S}_x, \hat{S}_y, \hat{S}_z]$ are the electron spin operators with eigenvectors $\{|\uparrow\rangle, |\downarrow\rangle\}$, $\vec{I}_i = [\hat{I}_{ix}, \hat{I}_{iy}, \hat{I}_{iz}]$ are the nuclear spin operators for nucleus $i \in 1, 2$ with eigenvectors $\{|\uparrow_i\rangle, |\downarrow_i\rangle\}$, A_1 and A_2 are the electron-nuclear hyperfine interactions, γ_e is the electron gyromagnetic ratio, γ_n is the nuclear gyromagnetic ratio, and B_0 is a static magnetic field oriented along the \hat{z} -axis. This Hamiltonian is commonly found in well-studied spin qubit systems such as the nitrogen-vacancy (NV) center in diamond [23], spins in quantum dots [24] and donors in silicon [12]. A Hamiltonian with this structure can be viewed as implementing the smallest possible parity check circuit for the Honeycomb Code, as shown in figure 9.

We take the specific example of a recent experiment where two ^{31}P nuclear spins were hyperfine-coupled to a common electron (2P1e) and entangled using a geometric CZ gate obtained through a rotation of the electron conditional on the state of the nuclei [13]. The gate fidelities extracted from GST were used in the preceding section to illustrate how gate errors build through a circuit. The numerical parameters of the Hamiltonian are $\gamma_e = -27.97 \text{ GHz/T}$, $\gamma_n = 17.23 \text{ MHz/T}$, $B_0 = 1.33 \text{ T}$, $A_1 = 95 \text{ MHz}$ and $A_2 = 9 \text{ MHz}$. The



large separation of energies in the system, whereby the electron Zeeman energy splitting $\gamma_e B_0 \approx 37.2$ GHz is orders of magnitude larger than both the nuclear Zeeman energies $\gamma_n B_0 \approx 22.8$ MHz and the hyperfine interactions, ensures that the eigenstates of the Hamiltonian are almost exactly the tensor products $\{|\uparrow\rangle, |\downarrow\rangle\} \otimes \{|\uparrow_1\rangle, |\downarrow_1\rangle\} \otimes \{|\uparrow_2\rangle, |\downarrow_2\rangle\}$, as shown in figure 9(b).

To implement the ZZ-SSPC gate we add the control Hamiltonian:

$$H_{rf}(t) = -\gamma_e \vec{B}_1 \vec{S} \cos(\omega t) - \gamma_n \vec{B}_1 \left(\vec{I}_1 + \vec{I}_2 \right) \cos(\omega t) \quad (17)$$

where B_1 is an the oscillating magnetic field strength oriented along \hat{y} . A ZZ-SSPC gate is obtained trivially by applying a bichromatic pulse of B_1 at the frequencies $\omega_{\uparrow\downarrow}$ and $\omega_{\downarrow\uparrow}$ corresponding to the resonance frequencies of the electron spin when the nuclei are in the odd-parity states $|\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle$. This is depicted with the yellow and green lines in figure 9(b). Calibrating the amplitude and duration of the pulse such that the electron undergoes an exact π -rotation implements the CNOT gates depicted in figure 6(c)(a). The two frequencies can be applied simultaneously because they both act on the same physical object (the electron). The SSPC gate is then followed by electron readout [25].

The XX-SSPC gate, shown in figure 5(c)(a), involves operations on both the electron and the nuclei, which require different time scales due to the vastly different gyromagnetic ratios. It is no longer immediately obvious that the whole operation can be executed in a single step. In order to verify whether an XX-SSPC gate exists, we adopt a widespread quantum control optimization algorithm known as gradient ascent pulse engineering (GRAPE) [26], which can be found within QuTip [27]. One of the reasons we chose to use GRAPE is because it enables us to account for real-life experimental limitations such as the maximum and minimum control amplitudes. Another reason is that the GRAPE algorithm was originally designed for NMR pulses [26, 28] and finds practical application in many experimental studies [29]. Notably, the results of the first experiment in QEC in 1998 [30] were greatly improved through the utilization of the GRAPE algorithm in 2011 [31]. The GRAPE algorithm in QuTip, is wrapped by the L-BFGS-B [32] optimization algorithm which takes into account second order derivatives and chooses the step size which is the change of position in the control landscape by itself. This helps the algorithm to use a smaller number of iterations. We utilized this algorithm in two distinct ways, namely modulated GRAPE and non-modulated GRAPE. The non-modulated GRAPE does not use the modulated control Hamiltonian, but it does use a control Hamiltonian with proper frequencies for both the electron and nucleus and control fields applied in two different directions (x and y). The modulated GRAPE, however, allows us to use a modulated control Hamiltonian, and we refer to this version of the algorithm as modulated GRAPE.

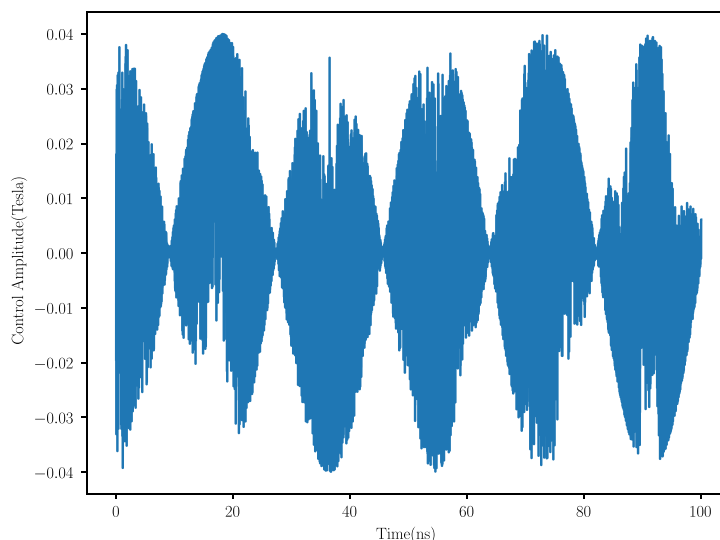


Figure 10. The first 100 ns of the optimised control pulses for the XX gate. The control amplitude represents the B_1 magnetic field applied to the spins.

For the modulated GRAPE method, we took into account the $\cos(\omega t)$ term and we modulated the control Hamiltonian with $\cos(\omega t)$ for each time slot with a proper angular frequency. As a first step, we set the maximum control amplitude (the B_1 magnetic field in our case) in the range of millitesla and we ran the algorithm for 1, 10, 100, 1000 μs evolution time for the XX gate. The number of time slots is one of the most important parameters of the algorithm. We need to provide a fine enough resolution to observe the oscillation of the pulse. Since we are in the laboratory frame, the time interval is determined by the highest frequency in the system, which in this case is the electron Larmor frequency. In a 1.33 T magnetic field, the electron precesses at 27.97 GHz, corresponding to a period of 35 ps. When the number of sampling per oscillation is taken into account, for each evolution time, the minimum time resolution is maintained at 10 picoseconds by allocating the appropriate number of time slots. To execute the algorithm with such a small time resolution, we utilized a high-performance computing cluster. As a result, we could have perfect gate fidelity for an evolution time of order 10 μs .

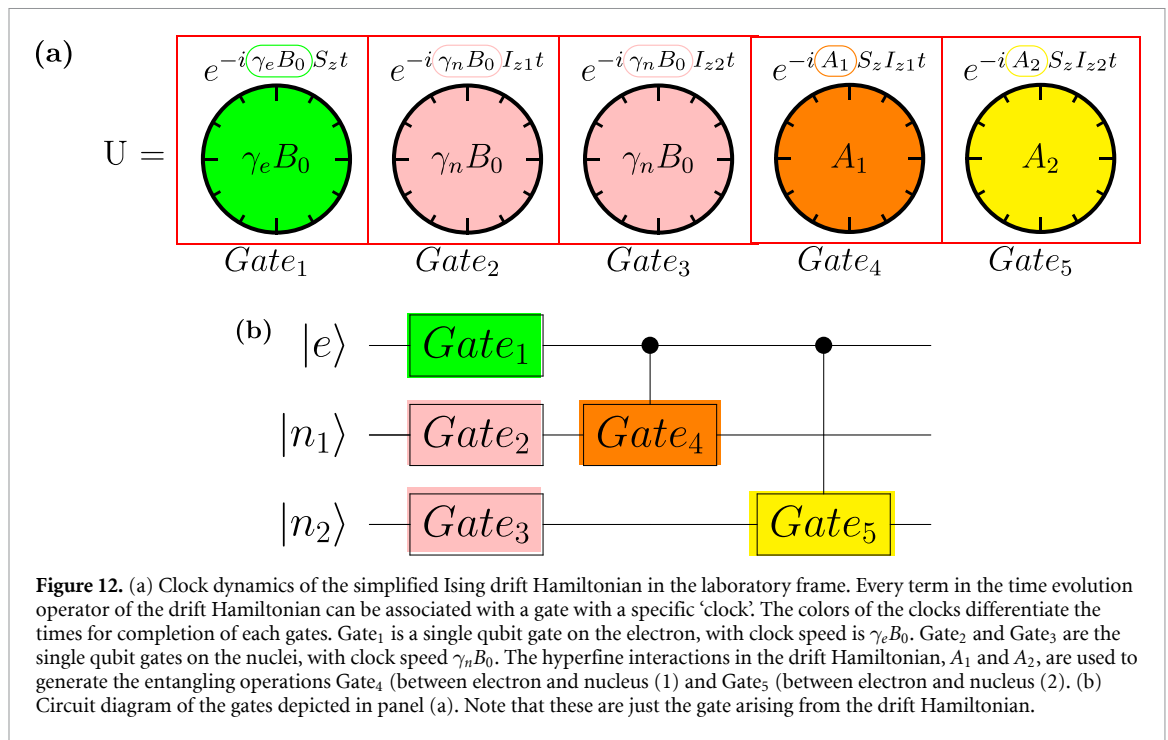
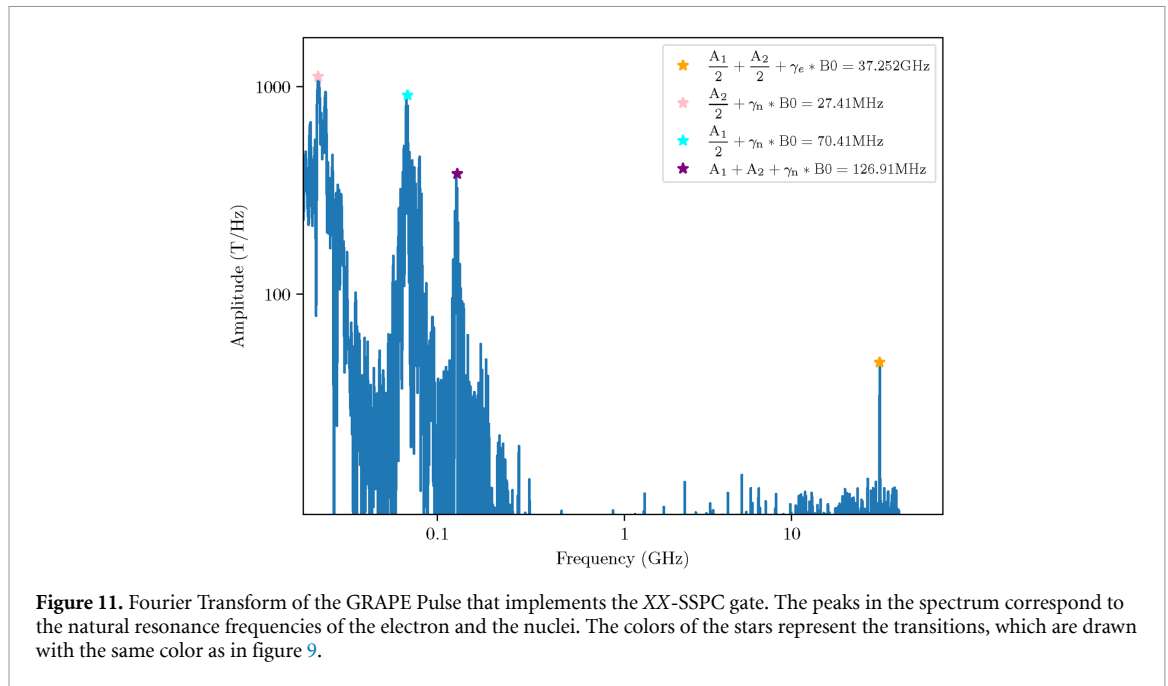
We then attempted to find a shorter gate evolution time, by increasing the control amplitude (still within experimentally feasible values). We set the evolution time to 4 μs for the XX gate. We set upper and lower bounds to 40 mT for the control field and we used 400 000 time slots which correspond to every 10 ps for a pulse. We run the algorithm for every frequency in the system and the algorithm was able to find XX parity check gates with a single pulse with an accuracy of 0.9999⁶. The first 100 ns of the control amplitudes are shown in figure 10.

To verify that the pulse found by GRAPE was physically meaningful, we then calculated its fast Fourier transform (FFT), shown in figure 10. Since the XX gate is an entangling gate formed through a combination of operations on both nuclei and electrons, we expect the spectrum of the pulse to consist of control signals at the natural resonance frequencies of the electron (ESR) and the nuclei (NMR). Indeed, the FFT in figure 11 reveals all the expected frequencies: electron spin resonance (orange marker in figure 11, corresponding to the orange line in figure 9(b)), and nuclear magnetic resonances (blue for nucleus 1, pink for nucleus 2). Interestingly, GRAPE also uses a stimulus at frequency $A_1 + A_2 + \gamma_n B_0$ which does not correspond to any natural resonance of the system (purple marker in figure 11).

We then tried to find a pulse for even shorter evolution times. The shortest evolution time that we could find a pulse for is 2 μs with a 0.98 fidelity with a modulated control Hamiltonian in the x direction. Note that, when we decreased the total evolution time from 4 μs to 2 μs , we also decreased the time resolution from 400 000 points to 200 000 points so that the time resolution stayed the same. Overall, GRAPE was consistently able to find one single pulse that can implement the gate without decomposing it into one- and two-qubit gates, and which does not require exceedingly long gate times while taking into account the experimental limitations on the maximum control amplitude.

To understand whether there are any advantages or disadvantages in terms of experimental feasibility of using the XX gate over the decomposed parity check circuit, we took a further step and attempted to

⁶ This is the theoretical fidelity and this demonstrates the feasibility of implementing this gate under the given conditions.



implement the decomposed parity check circuit, as defined in figure 5(a), by individually implementing each one- and two-qubit gate using GRAPE. Surprisingly, while we were able to implement the Hadamard gates within a proper gate evolution time of $0.5 \mu\text{s}$, the implementation of CNOT gates with high fidelity required a much longer time, around $100 \mu\text{s}$. As a result, the total timing for the decomposed parity check circuit fell in the ~ 100 microsecond range. This analysis shows that the system under study—two nuclear spins hyperfine-coupled to an electron—is naturally suited to implement a SSPC gate, thanks to the fact that each qubit has a separate resonance frequency, and that the ancilla (the electron) has resonances that each constitute a natural rotation conditional on the state of the two nuclei.

In the next part of this work, we analytically study a modified drift Hamiltonian to gain an understanding of the appropriate timescale for the SSPC gate and the decomposed parity check circuit. In pursuit of this goal, we performed a thorough investigation of the clock dynamics associated with the drift Hamiltonian in the laboratory frame (figure 12). To afford an analytical solution, we approximate the drift Hamiltonian as an Ising type modification to equation (16). While the result of GRAPE explicitly examines

the active pulsing required to realize the unitary for the SSPC gate without loss of generality, here we analytically examine the passive drift Hamiltonian for the Ising Hamiltonian in equation (18) (figure 12(a)).

$$H = -\gamma_e B_0 S_z - \gamma_n B_0 I_{z1} - \gamma_n B_0 I_{z2} + A_1 S_z I_{z1} + A_2 S_z I_{z2}. \quad (18)$$

The time evolution operator, $U = e^{iHt}$, can be expanded as

$$U = \underbrace{e^{-i(\gamma_e B_0) S_z t}}_{Gate_1} \underbrace{e^{-i(\gamma_n B_0) I_{z1} t}}_{Gate_2} \underbrace{e^{-i(\gamma_n B_0) I_{z2} t}}_{Gate_3} \underbrace{e^{i(A_1) S_z I_{z1} t}}_{Gate_4} \underbrace{e^{i(A_2) S_z I_{z2} t}}_{Gate_5} \quad (19)$$

Each of the exponential terms in equation (19), which are the free evolution times of the drift Hamiltonian, will generate a gate and every gate will have its own clock (figure 12(a)). These clocks correspond to the time taken for the desired gates. Gate₁, Gate₂, Gate₃, Gate₄ and Gate₅ are the natural gates which can be obtained from the evolution of the drift Hamiltonian (figure 12(b)). However, if we want to create specific gates, such as a CZ gate between an electron and the first nucleus using the hyperfine interaction denoted as A_1 , in other words, if we want Gate₄ to function as a CZ gate, then we need to have the control Hamiltonian and pulse the system. In this part of the study, our focus is on understanding the approximate timescale for the SSPC gate and the decomposed parity check circuit that were derived by using GRAPE. When considering the control Hamiltonian, the time required for the SSPC gate and the decomposed parity check circuit will change. However, the relative timescales between the SSPC gate and the decomposed parity check circuit will be comparable for analytical analysis and the numerical analysis. Since we are not using the control Hamiltonian, we do not refer to the gates as CZ gate or ZZ gate. Instead, we refer to them as CZ-like gate and ZZ-like gate. If we want to implement a decomposed ZZ parity check circuit as shown in figure 6(a), we need to implement CZ-like dynamics, since the ZZ parity check circuit can be implemented either by using the CZ gate with Hadamard gates or directly with two CNOT gates. However, if we want to implement an entangling gate between the electron and the first nucleus, we need to use the hyperfine coupling A_1 . But, the hyperfine coupling A_2 is also present and always on. Therefore, we need to find a time that allows us to perform an entangling gate (Gate₄) between the electron and the first nucleus and the identity (Gate₅) between the electron and the second nucleus, so that the second hyperfine interaction does not disturb the system. To find such a time, we need to consider the clock dynamics of the drift Hamiltonian. The first entangling gate will be between the electron and the first nucleus. A π rotation around the Bloch sphere will be a CZ rotation since $e^{i\pi} = -1$. The time for Gate₄ to function as a CZ-like gate will be $\pi/95\text{MHz} \approx 33.06\text{ns}$. For Gate₅ to be the identity operation, it should induce a 2π rotation, which takes $2\pi/9\text{MHz} \approx 698.13\text{ns}$. However, there will be some error associated with timing, denoted as t_{error} , as achieving the exact duration is not always possible. To find the appropriate time for both gates, we need to solve:

$$\begin{aligned} ((a \times t_{\text{Gate4}}) - (b \times t_{\text{Gate5}}))^2 &\leq t_{\text{error}}^2, a > 0, b > 0, \\ \text{where } a \text{ is an odd integer and } b \text{ is an integer} \end{aligned} \quad (20)$$

Here, the timing error, t_{error} is chosen as 0.01, and consequently, probability of experiencing a Pauli error is equal to the t_{error}^2 is 0.0001. In this situation, the evolution of Gate₄ should at least make $a = 6981 \pi$ rotations while the evolution of Gate₅ should at least make $b = 331 2\pi$ rotations so that we can have a perfect CZ-like gate. The total time for a perfect CZ-like gate between the electron and the first nuclei is $33.1 * 6981 = 231.07 \mu\text{s}$

Now, if we want to make a perfect CZ-like gate between the electron and the second nucleus, one π rotation for Gate₅ to function as a CZ-like is equal to $\pi/9 \text{ MHz} \approx 349.06\text{ns}$ and the time for one 2π rotation for Gate₄ to function as identity is equal to $\approx 66.13 \text{ ns}$. If we solve:

$$\begin{aligned} ((a \times t_{\text{Gate5}}) - (b \times t_{\text{Gate4}}))^2 &\leq t_{\text{error}}^2, a > 0, b > 0, \\ \text{where } a \text{ is an odd integer and } b \text{ is an integer} \end{aligned} \quad (21)$$

the fastest possible time will occur when we take a single π rotation time for Gate₅ as 349.06 ns and single 2π rotation time for Gate₄ as 66.13 ns. In this case, $a = 887$ is the number of π rotations for the evolution of Gate₅. Correspondingly, the total time for a clear CZ-like gate for the electron and the second nucleus is equal to $349.06 \times 887 = 309.06 \mu\text{s}$. As a result, if we want to implement the circuit which is defined in figure 6(a), the simplified Hamiltonian can naturally achieve that in a range of hundreds μs . In the second part of this analytical calculation, we will analyze the drift Hamiltonian for the ZZ-like gate. For

implementing such a gate, both the Gate₄ and the Gate₅ should function as CZ gates so they should make a π rotation. For Gate₄ to function as CZ-like, the time for one single π rotation is $\pi/95$ MHz = 33.06ns and for Gate₅, the time for one single π rotation is $\pi/9$ MHz = 349.06 ns. To see if our drift Hamiltonian can have a suitable time for such a gate, we need to solve equation (22).

$$\begin{aligned} ((a \times t_{\text{Gate4}}) - (b \times t_{\text{Gate5}}))^2 &\leq t_{\text{error}}^2, a > 0, b > 0, \\ \text{where both } a \text{ and } b \text{ are an odd integer.} \end{aligned} \quad (22)$$

This yields $a = 1805$ and $b = 171$ and total time for the desired unitary of $\approx 60 \mu\text{s}$. As a result, implementing two separated CZ-like dynamics requires $\approx 540.13 \mu\text{s}$ in total whereas implementing a ZZ-like dynamics requires $\approx 60 \mu\text{s}$ which is 9 times faster than implementing a traditional parity check circuit. This explanation clarifies why the GRAPE algorithm discovered the SSPC gate approximately 10 times faster than the decomposed parity check circuit. This is because the SSPC gate does not suffer from imbalanced hyperfine interaction, whereas the isolated entangling gate does. The codes concerning this section, encompassing both the XX gate and the decomposed parity check gate, can be found in [33–35].

5. Conclusions

In this theory work, we developed a workflow which calculates the perfection rate of the whole parity check with given experimental data. We numerically showed how errors are accumulating and causing a decrease of the perfection rate of the parity check circuit with current sources (with single and two qubit gate and M_{pp1}) from the experimental GST matrices. Our findings indicate that relying solely on the fidelity of 1- and 2-qubit gate operations is inadequate in assessing the effectiveness of constructing the fundamental elements of QEC, specifically the parity check circuits. Then, we introduced new SSPC unitaries which allow us to make parity check measurements in one step. We modified the universal gate set such that the new unitaries are the most natural gate set for QEC. We compared the traditional parity check gate (1- and 2-qubits gates) and the SSPC gate for an XX parity check scenario in the presence of noise.

We provided an experimentally realizable example based on spin qubits in silicon, and used the GRAPE algorithm to find a single pulse capable of implementing the SSPC gate within experimental limitations. Furthermore, we implemented the decomposed parity check circuit using GRAPE. We found that the SSPC gate is naturally suited for this system, and outperforms a decomposed gate.

Many have asked what the ‘killer-app’ is for a large-scale, error-corrected quantum computer, this is error-correction. The vast majority of the computation occurring in a large-scale machine is dedicated to correcting its own errors. As such, fundamental gate sets used at the physical level should be targeted at this application. We have introduced a complete workflow that allows experimentalists to characterise more complex gate sets and connect outputs of characterisation experiments to commonly used models used in QEC analysis. This allows quantum hardware engineers to potentially redesign and test fundamental gate libraries, increase performance, and accelerate the realisation of fault-tolerant, error-corrected quantum computers

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://zenodo.org/doi/10.5281/zenodo.10795552>.

Acknowledgment

The work was funded by the Australian Research Council (CE170100012). G Ü acknowledges support from the Sydney Quantum Academy. This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government. SJD was supported with funding from the Defense Advanced Research Projects Agency [under the Quantum Benchmarking (QB) Program Under Award Nos. HR00112230007 and HR001121S0026 contracts]. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. We thank S Asaad, M Mađzik and R Savvitskyy for sharing GST experimental results and useful discussions. We thank B C Sanders for discussions on quantum control and optimal pulse methods, and A Pitchford for support on the GRAPE algorithm.

Appendix. Workflow—understanding the current way of constructing parity check circuit and its drawback in the context of QEC

The current standard method for constructing multi-body measurements involves applying controlled-gates, like the CNOT (CX) gate for X parity measurement, between data qubits and ancilla qubits, followed by a one-body measurement on the ancilla qubit. The ancilla is crucial in this process, as without it, the measurement result will not provide us with the parity information. While this works in theory, it is expensive in the lab since each layer of gates added results in a decrease in the total perfection of the quantum circuit. Moreover, we may be able to do better as natural two qubits measurements are possible in some systems. Here, we provide details on how to derive the total perfection rate of the circuit using stochastic Pauli channels. This calculation assumes that we apply each of the operators just before making the measurement and model every imperfect gate as perfect gates plus some error.

A.1. Preliminaries

- (i) Superoperators represent a linear map between initial and final density matrices of the system. They are the mathematical representations which bring one density matrix to another. The unitary evolution of a density matrix is defined by equation (A.1) where U is the unitary matrix and its dimension is $2^n \times 2^n$, n is the number of qubits, ρ_i is the initial state of the system and ρ_f is the final state of the system after this unitary is applied.

$$\rho_f = \mathcal{E}(\rho_i) = U\rho_i U^\dagger \quad (\text{A.1})$$

where \mathcal{E} describes a quantum channel.

Now, there exists a linear representation of \mathcal{E} due to Choi–Jamakowski which brings $|\rho_i\rangle\rangle \mapsto |\rho_f\rangle\rangle$ and its representation based on the vectorization of the density matrix with respect to the column basis: [36].

$$U \otimes U^\dagger |\rho_i\rangle\rangle = |\rho_f\rangle\rangle. \quad (\text{A.2})$$

Dimensions of the Superoperators are equal to $2^{2n} \times 2^{2n}$ where n is the number of qubits and $|\cdot\rangle\rangle$ represents vectorization.

- (ii) Equation (A.1) is the special case where we have applied the unitary U in the absence of environmental noise. However, in the presence of imperfect gates or environmental noise, equation (A.1) will not be enough to represent the imperfections. In this case, we will need a more general mapping to represent our system. Kraus operators are the most general representation of our system in the presence of noise and they represent both the systematic gate errors and the errors induced by environmental decoherence [14]. Then, the density matrix can be written as

$$\rho_f = \mathcal{E}(\rho_i) = \sum_{k=1}^N A_k \rho_i A_k^\dagger, \quad (\text{A.3})$$

with

$$\sum_{k=1}^N A_k^\dagger A_k = 1, \quad (\text{A.4})$$

where $N \leq 2^{2n}$ and $\{A_k\}$ are the set of Kraus operators which replace the unitary in equation (A.1). The Kraus operators, $\{A_k\}$, are not necessarily unitary operators, hermitian operators or invertible. If $N = 1$, this means that we have the ideal case and this brings us back to the equation (A.1) where A_k is U . Equation (A.4) is called the completeness condition if the map is trace-positive, TP. Note that since the quantum channels considered here map the system to itself, we only consider Kraus operators to be square matrices.

- (iii) χ —Process matrix

When we write Kraus operators in the Pauli basis, in other words, A_k is extended in terms of Pauli operators, then we will have $A_k = \sum_{j=1}^{d^2} a_{jk} P_j$ where $P_j \in P^{\otimes n}$ and $P = \{I, X, Y, Z\}$. If we write that in equation (A.3), then we will have:

$$\rho_f = \mathcal{E}(\rho_i) = \sum_{j=1}^{d^2} \chi_{jk} P_j \rho_i P_k \quad (\text{A.5})$$

where $\chi_{jk} = \sum_i a_{ij} a_{ik}^*$. The χ matrix is a complex-valued matrix with dimension $2^{2n} \times 2^{2n}$ and is a complete map of ρ_f . The χ matrix is also the output of process tomography [21].

- (iv) The Pauli transfer matrix (PTM), which is shown as R in literature [21, 36–38], is another useful representation of a quantum channel and an outcome of GST [21, 22]. It is defined by the vectorization of a quantum channel in the Pauli basis and it can be written as:

$$(R_\varepsilon)_{ij} = \frac{1}{d} \text{Tr} \{ P_i \mathcal{E} (P_j) \}. \quad (\text{A.6})$$

Here, P_j 's represents ordering in the strings of Pauli operators, i.e. $\{I, X, Y, Z\}$. The PTM, R , is the special case of Superoperators where the vectorization is made in the Pauli basis. One notable advantage of using the PTM, R , is its exclusive utilization of real elements. Furthermore, R facilitates the straightforward determination of whether a quantum operation is trace-preserving or unital. It is also noteworthy that for any Clifford operation, each row and column of R with unit magnitude contains a single non-zero element. The PTM makes it easy to evaluate the result of multiple gates acting in succession. The process matrix does not have this property. This is why the output matrices of GST are PTM [22].

- (v) The Density matrix in the context of QEC:

We shall commence the construction of the density matrix for a single qubit. For the density matrix of multi-qubit systems, we refer the reader to the appendices A.5 and A.6). In this case, the dimensions of Kraus operators will be 2×2 . Once we found the Kraus operators, they can be written in terms of Pauli matrices as in equation (A.7), since every complex 2×2 matrix can be written by Pauli Matrices⁷.

$$A_k = C_0 I + C_1 X + C_2 Y + C_3 Z. \quad (\text{A.7})$$

If the Kraus operators are in equation (A.3) are decomposed as in equation (A.7), then we have:

$$\rho_f = \sum_{a \in M} \sum_{b \in M} a \rho_i b^\dagger \quad (\text{A.8})$$

where $M = \{C_0 I, C_1 X, C_2 Y, C_3 Z\}$.

Suppose the map described by equation (A.8) acts on a qubit that is part of a larger QEC-encoded system. We will now explore the observable effect of this mapping when we run a QEC cycle. For this purpose, let us consider an error model, such as the one given in equation (A.9):

$$\rho_f = p_i \rho + p_x X \rho X + p_{xy} X \rho Y. \quad (\text{A.9})$$

Here ρ represents a qubit, while X and Y represent the physical error on the qubits in the code block. An ancilla block, represented by the density matrix ρ_0^E is coupled to the system and the QEC unitary which is called U_{QEC} is run:

$$U_{\text{QEC}} (\rho_f \otimes \rho_0^E) U_{\text{QEC}}^\dagger = p_i \rho \otimes |E\rangle\langle E| + p_x X \rho X \otimes |E_x\rangle\langle E_x| + p_{xy} X \rho Y \otimes |E_x\rangle\langle E_y| \quad (\text{A.10})$$

Here, $|E\rangle$, $|E_x\rangle$ and $|E_y\rangle$ represent the three orthogonal syndrome states of the ancilla qubit and they are used the detect errors on the qubits. Now, when the ancilla qubit is measured, the system will be collapsed to one of the two states in equation (A.11)

$$\rho_f \Rightarrow \frac{\langle E | \rho | E \rangle |E\rangle\langle E|}{\text{Tr}(\rho |E\rangle\langle E|)} \text{ and } \rho_f \Rightarrow \frac{\langle E_x | \rho | E_x \rangle |E_x\rangle\langle E_x|}{\text{Tr}(\rho |E_x\rangle\langle E_x|)} \quad (\text{A.11})$$

The off-diagonal term, in this case the $p_{xy} X \rho Y$ term in the error model, is never observed. After the measurement of the ancilla, we have two possible states which are shown in equation (A.12)

$$\begin{aligned} &\rho \otimes |E\rangle\langle E| \text{ with probability } p_i \\ &X \rho X \otimes |E_x\rangle\langle E_x| \text{ with probability } p_x \end{aligned} \quad (\text{A.12})$$

where $p_i + p_x = 1$. The off-diagonal term is eliminated because we are about to make a measurement in the E, E_x basis and the measurement result will collapse one of the diagonal terms. Although it was

⁷ Also every $2^n \times 2^n$ matrix can be written as a linear combination of the tensor products of n Pauli matrices. The total number of independent tensor products of Pauli matrices is 2^{2n} .

written in [14], it is once more worth noting that not only the off-diagonal term is eliminated but the final density matrix is collapsed to clean codeword states with bit-flip errors. Here, we do not use the twirling method which is a way to convert the off-diagonal terms of the matrix into the diagonal terms [39, 40]. In quantum computation, there are some protocols that use twirling, such as purification protocols [41–43]. However, when we write errors in terms of stochastic Pauli channels, we let unitaries evolve until the measurement is done. This is why, when we calculate the effective errors in the context of QEC, we only care about diagonal elements and we ignore off-diagonal elements. Hence, we are finding the diagonal elements of the χ matrix and we are writing the Kraus operators in terms of these diagonal elements of the χ matrix. It is true that we are losing the information by ignoring the off-diagonal terms. In equation (A.8), once cancelling the off-diagonal elements, we have,

$$\rho_f = p_i I \rho_i I + p_x X \rho_i X + p_y Y \rho_i Y + p_z Z \rho_i Z \quad (\text{A.13})$$

where $p_i = (|C_0\rangle)^2$, $p_x = (|C_1\rangle)^2$, $p_y = (|C_2\rangle)^2$, $p_z = (|C_3\rangle)^2$. This formula represents effective errors in the context of QEC for 1 qubit. As an illustration, in the case of implementing the identity gate, the coefficient p_i serves as an indicator of the gate's efficacy. A value of $p_i = 1$ signifies the perfect implementation of the identity gate. The equation (A.13), can be written for any state and in QEC, the errors in the quantum channel are analyzed after the application of the desired unitary. Because of that, to utilize the equation (A.13) in a more general way, the ρ_i is replaced with ρ_g where $\rho_g = U \rho_i U^\dagger$ and the U is the desired unitary that we want to implement. Then, the equation (A.13), becomes equation (A.14):

$$\rho_f = p_i I \rho_g I + p_x X \rho_g X + p_y Y \rho_g Y + p_z Z \rho_g Z \quad (\text{A.14})$$

In equation (A.14), the p_i represents the probability of having no error. We define it as the perfection rate. If $p_i = 1$, we are in the ideal case with no error and we turn back to equation (A.1). For all the other values of p_i , we use the Kraus decomposition. The coefficients p_x , p_y and p_z are, now, the probability of having X errors, the probability of having Y errors and the probability of having Z errors, respectively. We start with the experimental data, in our case the experimental data is the PTM matrix, and we end up with equation (A.14). This result is the important conclusion of the section. If we want to determine error rates subsequent to the implementation of a unitary operation within the framework of QEC, every operation performed in the laboratory will be denoted as equation (A.14). This equation represents a whole process and the coefficient p_i in the equation represents the perfection of the operation that cannot be written simply as the fidelity. For example, if we have 99% fidelity for a specific operation, this does not mean that the coefficient $p_i = 0.99$. To find the coefficient p_i , we need to do the whole workflow: we first take the experimental process matrices of the system and find the Kraus operators. Subsequently, we then write the Kraus operators in terms of stochastic Pauli channels and determine the value of coefficient p_i , which is the perfection rate and it is not equal to fidelity. It is less than the fidelity. Neither the fidelity means the perfection rate, nor the infidelity means the error rate. To finding the error rate, we need information beyond the fidelity [44] A.4. We give a complete tutorial below A.4 regarding this process. If we take figure A1 as an example and calculate the probability of experiencing certain errors when gates are applied to qubits, the equation which represents the perfection of the operator will be equation (A.15). The detailed derivation of equation (A.15) is shown in appendix A.2.

$$P_I = p_{i2} p_{i1} + p_{x2} p_{x1} + p_{y2} p_{y1} + p_{z2} p_{z1}. \quad (\text{A.15})$$

The value P_I is the perfection rate for the whole circuit and it represents the final succession rate after all of the accumulated errors in the widget. The important point is, that P_I is not just the multiplication of perfection rates p_i . The cancellation terms also contribute to P_I since $X \otimes X = I$, $Y \otimes Y = I$ and $Z \otimes Z = I$. We will call P_I as the perfection rate of the widget.

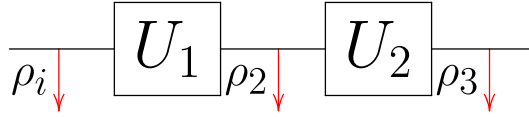


Figure A1. A one qubit circuit example which will be used to calculate effective errors in the context of QEC. At the beginning, the circuit will have ρ_i . After applying a U_1 gate, the density matrix will become ρ_2 . The final density matrix will be ρ_3 , which can be written as ρ_f after applying U_3 .

A.2. Calculating the final density matrix in the context of QEC

We explicitly write the density matrices shown in figure A1. Let's start with writing ρ_2

$$\rho_2 = p_{i1}IU_1\rho_iU_1^\dagger I + p_{x1}XU_1\rho_iU_1^\dagger X + p_{y1}YU_1\rho_iU_1^\dagger Y + p_{z1}ZU_1\rho_iU_1^\dagger Z \quad (\text{A.16})$$

where p_{i1} is the perfection rate of U_1 gate. p_{x1} , p_{y1} , and p_{z1} respectively represent the probabilities of having X, Y, and Z errors in the channel after applying the unitary U_1 . The system now in ρ_2 . Then we apply U_2 gates and the system becomes ρ_3 .

$$\rho_3 = p_{i2}IU_2\rho_2U_2^\dagger I + p_{x2}XU_2\rho_2U_2^\dagger X + p_{y2}YU_2\rho_2U_2^\dagger Y + p_{z2}ZU_2\rho_2U_2^\dagger Z \quad (\text{A.17})$$

where p_{i2} is the perfection of U_2 gate which. p_{x2} , p_{y2} and p_{z2} respectively represent the probabilities of having X, Y, and Z errors in the channel after applying the unitary U_2 . If we put equation (A.16) in equation (A.17), then we will have:

$$\begin{aligned} \rho_3 = & \overline{p_{i2}}IU_2[\overline{p_{i1}}IU_1\rho_iU_1^\dagger I + p_{x1}XU_1\rho_iU_1^\dagger X + p_{y1}YU_1\rho_iU_1^\dagger Y + p_{z1}ZU_1\rho_iU_1^\dagger Z]U_2^\dagger I + \\ & \overline{p_{x2}}XU_2[p_{i1}IU_1\rho_iU_1^\dagger I + \overline{p_{x1}}XU_1\rho_iU_1^\dagger X + p_{y1}YU_1\rho_iU_1^\dagger Y + p_{z1}ZU_1\rho_iU_1^\dagger Z]U_2^\dagger X + \\ & \overline{p_{y2}}YU_2[p_{i1}IU_1\rho_iU_1^\dagger I + p_{x1}XU_1\rho_iU_1^\dagger X + \overline{p_{y1}}YU_1\rho_iU_1^\dagger Y + p_{z1}ZU_1\rho_iU_1^\dagger Z]U_2^\dagger Y + \\ & \overline{p_{z2}}ZU_2[p_{i1}IU_1\rho_iU_1^\dagger I + p_{x1}XU_1\rho_iU_1^\dagger X + p_{y1}YU_1\rho_iU_1^\dagger Y + \overline{p_{z1}}ZU_1\rho_iU_1^\dagger Z]U_2^\dagger Z \end{aligned} \quad (\text{A.18})$$

Now, the coefficients are inside the dashed lines gives us to total perfection of the circuit after applying operations. As a result we get equation (A.19):

$$P_I = p_{i2}p_{i1} + p_{x2}p_{x1} + p_{y2}p_{y1} + p_{z2}p_{z1} \quad (\text{A.19})$$

where P_I is the perfection rate of the widget after applying all the unitaries in the quantum circuit. For the accumulated errors, for instance the accumulated x errors, we will have the equation (A.20):

$$P_X = p_{i2}p_{x1} + p_{x2}p_{i1} + p_{y2}p_{z1} + p_{z2}p_{y1} \quad (\text{A.20})$$

If we had only one step, we would have ρ_2 exclusively. In that case, P_I would be equal to p_{i1} , and the X error would be represented by $p_x = p_{x1}$. However, as the circuit depth increases, the occurrences of X, Y, and Z errors also increase, leading to a decrease in the fidelity P_I . In equation (A.19), it is important to note that $P_I < p_{i1}$ since errors accumulate within the circuit. Below, we will explain our algorithms for finding the perfection rate:

A.3. The algorithm

The algorithm receives the experimental data as its input, and subsequently delivers the two key performance indicators for the quantum circuit: the perfection rate and the total accumulated error rate. In the interest of convenience, our algorithm has been decomposed into three distinct phases:

- Finding Kraus Operators from GST:
- Finding Errors in the Context of QEC
- Calculating the perfection rate of the whole quantum circuit

The initial stage of the algorithm involves specifying the type of the matrices. In case of GST, the suitable format is the PTM (A.6) and from the GST experiment, $\pi/2$ and CNOT gate matrices are constructed and analysed. The important thing here is to correctly classify the matrix type. This can be achieved through the use of libraries such as Qiskit [45] and Qutip [46]. If we do not explicitly specify the matrix type in our code and attempt to analyze the matrix, for example, to check its physicality (that is CPTP: completely positive and trace preserving) using some libraries, directly from the array, we will receive a False output because we did not specify the matrix type. In addition, if we misidentify the matrix type, for example, when the matrix type is a PTM but we mistakenly identify it as a χ matrix in our code, even though the χ matrix is CPTP, we may receive incorrect results when checking the physicality of the matrix due to the incorrect matrix type specification.

In the first phase of the algorithm, we first check if the matrix is completely positive (CP). If it is not a CP matrix, this means that the experimental method used gives a different result than the PTM. In this case, it is recommended to check the output of the experimental method used in the lab, as it may yield different types of output, such as χ matrix. After specifying the type of the matrix correctly, the second step in Phase 1 is to find the Kraus operators. Then we check if we can find the Kraus operators correctly by using the completeness condition equation (A.4) and see if this gives us the identity matrix. If the completeness condition gives us the identity matrix, then we satisfy the trace-preserving condition for the quantum channel. After finding the Kraus operators, the second step is to calculate errors in the context of QEC.

For finding errors, we write each Kraus operator in terms of Pauli matrices as shown in Phase 2 and then the density matrix was written in terms of effective errors in the context of QEC as in equation (A.40).

In the third phase of the algorithm, since we have 3 qubits (2 data qubits and 1 ancilla qubit), all triplet combinations of Pauli matrices should be considered as possible error probabilities and we take into account all the diagonal terms. In this case, we have 64 different coefficients for the density matrix. The perfection value is calculated in equation (A.41) in terms of accumulating errors. Here, each density matrix includes the previous density matrix which has 64 coefficients. Since we have 6 steps in the XX parity check circuit, then at 6th step, we will have 64^6 coefficients and from those 64^6 , once we extracted the coefficient for the dominated term, equation (A.41) was found. Last phase of the algorithm shows how to calculate the perfection rate for XX parity check circuit. The complete code for replicating the results presented in figure 4, as well as the software for analyzing errors within the context of QEC, can be found in [35, 47].

Error analysis in the context of QEC.

Phase 1: Preparation—Finding Kraus Operators from GST

```

1: Input : Ideal and experimental GST Matrices
2: Output : Kraus List
3:  $M_{\text{full}} = M_{\text{GST}} * M_{\text{GSTerror}}$ 
4:  $M_{\text{PTM}} = \text{PTM}(M_{\text{full}})$ 
5: if  $M_{\text{PTM}}$  is Complete Positive matrix == True then
6:   list( $M_{\text{Kraus}}$ ) = Kraus( $M_{\text{PTM}}$ )
7:   if sum(i), for i in  $M_{\text{Kraus}}$  == I then
8:     return list( $M_{\text{Kraus}}$ )
9:   end if
10: end if

```

Algorithm. Error Analysis in the Context of QEC.

Phase 2: Finding Errors in the context of QEC

```

1: Input : Kraus List
2: Output : Errors
3:  $i$ =Pauli0, $x$ =Pauli1, $y$ =Pauli2, $z$ =Pauli3,result=[], $N$ =qubit number
4: procedure COEFFICIENTS(list( $M_{Kraus}$ ), $N$ )
5:   paulis = list(product([ $i,x,y,z$ ],repeat= $N$ ))
6:   spaulis = list(product(["i","x","y","z"],repeat= $N$ ))
7:   for kindex,kraus in enumerate(krauslist) do:
8:     for index,pauli in enumerate(paulis) do:
9:       error =  $\frac{1}{2^N}$  Trace((kraus* (TensorProduct(pauli[0],...,pauli[N])))
10:    end for
11:  end for
12:  return result
13: end procedure
14:  $d$  = defaultdict(complex)
15: for value, name, order in result do:
16:    $d$ [name] += abs(value) * abs(value)
17: end for

```

Algorithm. Error analysis in the context of QEC.

Phase 3: Computing the Perfection Rate

```

1: Input :  $H$  = errors for  $H$  gate in dictionary format,
2:    $I$  = errors for  $I$  gate in dictionary format,
3:   cnot = errors for CNOT gate in dictionary format
4:   meas = errors for measurement in dictionary format
5: Output : Perfection Rate
6: # For the first part of 2 body parity check circuit which corresponds  $H \otimes I \otimes I$ 
7:  $HII$  = defaultdict(complex)
8: for nameH, valueH in  $H$ .items() do:
9:   for nameI, valueI in  $I$ .items() do:
10:    for nameI, valueI in  $I$ .items() do:
11:     name = (*nameH *nameI *nameI)
12:      $HII$ [name] = valueH *valueI *valueI
13:    end for
14:   end for
15: end for
16: # For the second part of 2 body parity check circuit which corresponds
     $CNOT \otimes I \otimes I$ 
17:  $CNOTI$  = defaultdict(complex)
18: for nameCNOT, valuecnot in cnot.items() do:
19:   for nameI, valueI in  $I$ .items() do:
20:    name = (*nameCNOT *nameI)
21:     $CNOTI$ [name] = valuecnot *valueI
22:   end for
23: end for
24: # For the third part of 2 body parity check circuit which corresponds non-adjacent CNOT
25:  $ICNOT$  = defaultdict(complex)
26: for nameI, valueI in  $I$ .items() do:
27:   for nameCNOT, valuecnot in cnot.items() do:
28:    name = (*nameI *nameCNOT)
29:     $ICNOT$ [name] = valueI * valuecnot
30:   end for
31: end for
32: # For the last part of 2 body parity check circuit which corresponds measurement gate
33:  $measII$  = defaultdict(complex)
34: for nameMeas, valueMeas in meas.items() do:
35:   for nameI, valueI in  $I$ .items() do:
36:     for nameI, valueI in  $I$ .items() do:
37:      name = (*nameMeas *nameI *nameI)
38:       $measII$ [name] = valueMeas *valueI *valueI
39:     end for
40:   end for
41: end for
42: # Calculating Perfection rate at each step
43: perfection rate = sum( $HII$ [ $k$ ]* $CNOTI$ [ $k$ ]* $ICNOT$ [ $k$ ]*
44:  $HII$ [ $k$ ]* $measII$ [ $k$ ] for  $k$  in  $HII$ )

```

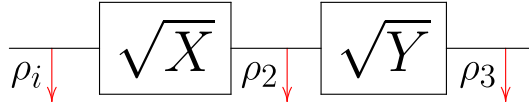


Figure A2. One qubit circuit example which will be used to calculate effective errors in the context of QEC. At the beginning, the circuit will have the ρ_i , after applying \sqrt{X} gate, the density matrix will become ρ_2 and the final density matrix will be ρ_3 after applying \sqrt{Y} .

A.4. Tutorial

Here, we will demonstrate the complete process for a single qubit. We take the circuit in figure A2 as an example:

We take the data from [48]. The average fidelity of the single qubit operation is 98.4%. The experimental GST matrix for \sqrt{X} gate is (A.21).

$$\sqrt{X}_{\text{GST-exp}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 8.86 \times 10^{-4} & 0.9864 & 0.01961 & 0.04048 \\ 0.01433 & 0.01039 & 0.01856 & -0.957 \\ -0.02782 & -0.03123 & 0.9487 & 0.008478 \end{pmatrix}. \quad (\text{A.21})$$

The ideal GST matrix for \sqrt{X} gate is (A.22).

$$\sqrt{X}_{\text{GST-ideal}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (\text{A.22})$$

Now the complete GST matrix for \sqrt{X} gate will be (A.23)

$$\sqrt{X}_{\text{GST}} = \sqrt{X}_{\text{GST-ideal}}^\dagger \times \sqrt{X}_{\text{GST-exp}} \quad (\text{A.23})$$

$$\sqrt{X}_{\text{GST}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 8.86 \times 10^{-4} & 0.9864 & 0.01961 & 0.04048 \\ -0.02782 & -0.03123 & 0.9487 & 0.008478 \\ -0.01433 & -0.01039 & -0.01856 & 0.9569 \end{pmatrix}. \quad (\text{A.24})$$

We then specify the matrix type in equation (A.24) as PTM [21, 36, 45] and then, we find the Kraus Operators for \sqrt{X} gate:

$$A_{k1} = \begin{pmatrix} -0.9828 - i1.522 \times 10^{-17} & 0.01241 - i0.01432 \\ -0.01333 + i5.944 \times 10^{-5} & -0.9898 + i0.02618 \end{pmatrix} \quad (\text{A.25})$$

$$A_{k2} = \begin{pmatrix} 0.07268 + i2.130 \times 10^{-16} & 0.09659 - i0.06736 \\ 0.1050 - i0.1321 & -0.07130 + i0.004211 \end{pmatrix} \quad (\text{A.26})$$

$$A_{k3} = \begin{pmatrix} 9.315 \times 10^{-3} + i4.151 \times 10^{-16} & -2.728 \times 10^{-3} + i1.084 \times 10^{-2} \\ -5.364 \times 10^{-3} - i1.597 \times 10^{-3} & -9.358 \times 10^{-3} + i3.656 \times 10^{-4} \end{pmatrix} \quad (\text{A.27})$$

$$A_{k4} = \begin{pmatrix} 1.140 \times 10^{-3} + i1.923 \times 10^{-12} & -4.041 \times 10^{-4} - i1.773 \times 10^{-3} \\ -3.195 \times 10^{-6} + i1.833 \times 10^{-3} & -1.112 \times 10^{-3} - i2.334 \times 10^{-5} \end{pmatrix}. \quad (\text{A.28})$$

To verify whether we have identified the correct Kraus operators, we need to apply equation (A.4). After applying equation (A.4), if we obtain the identity matrix, it indicates that the Kraus operators we have

determined are correct. The subsequent step involves expanding each Kraus operator in terms of Pauli matrices. Therefore, we will have the following expressions:

$$A_{k1} = (-0.9863 + i0.01309)I + (-0.00046 - i0.00713)X + (0.00719 + i0.01287)Y + (0.0035 - i0.01309)Z \tag{A.29}$$

$$A_{k2} = (0.00069 + i0.00211)I + (0.10081 - i0.09971)X + (-0.03236 - i0.00422)Y + (0.07199 - i0.00211)Z \tag{A.30}$$

$$A_{k3} = (-2 \times 10^{-5} + i0.00018)I + (-0.00405 + i0.00462)X + (-0.00622 + i0.00132)Y + (0.00934 - i0.00018)Z \tag{A.31}$$

$$A_{k4} = (1 \times 10^{-5} - i1 \times 10^{-5})I + (-0.0002 + i3 \times 10^{-5})X + (0.0018 - i0.0002)Y + (0.00113 + i1 \times 10^{-5})Z \tag{A.32}$$

To obtain the total final density matrix for the \sqrt{X} gate, we apply equation (A.8). We square the absolute value of each coefficient and then sum the coefficients based on their respective groups. As a result, we obtain equation (A.33).

$$\rho_2 = \rho_{\sqrt{X}} = (0.9730)I + (0.02019)X\rho_iX + (0.001325)Y\rho_iY + (0.005458)Z\rho_iZ \tag{A.33}$$

As seen from equation (A.33), the value of p_i is 0.973, which is not equal to the average fidelity of the 1Q gate operation, which is 98.4%. Next, we perform the same process for the \sqrt{Y} gate:

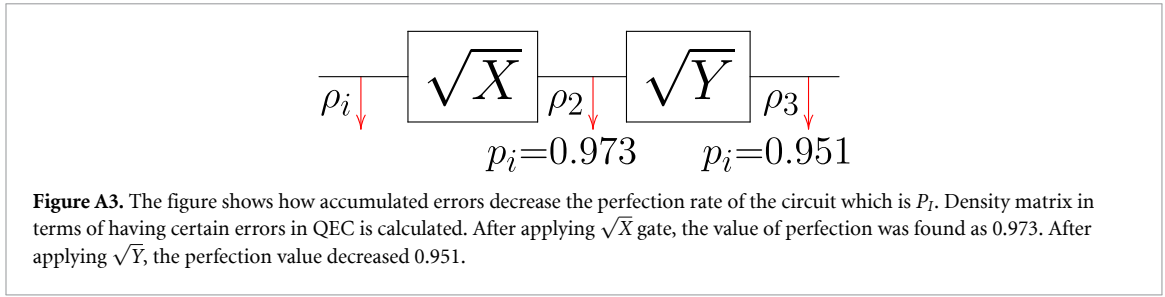
$$\sqrt{Y}_{\text{GST-exp}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -0.00995 & 0.03178 & 0.03606 & 0.9617 \\ -0.001533 & 0.04237 & 0.9806 & -0.01486 \\ -0.00903 & -0.9692 & 0.008547 & 0.01403 \end{pmatrix}. \tag{A.34}$$

The ideal GST matrix for \sqrt{Y} gate is (A.35).

$$\sqrt{Y}_{\text{GST-ideal}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}. \tag{A.35}$$

The final density matrix for \sqrt{Y} gate is found as equation (A.36).

$$\rho_3 = \rho_{\sqrt{Y}} = (0.9779)I + (0.006719)X\rho_2X + (0.01241)Y\rho_2Y + (0.002998)Z\rho_2Z \tag{A.36}$$



As it seen from equation (A.36), ρ_3 has ρ_2 . To calculate final coefficients of ρ_3 , we put the equation (A.33) in equation (A.36) as it shown in (A1). Then we calculate the P_I coefficient by the equation (A.19).

$$P_I = (0.9779) \times (0.973) + (0.006719) \times (0.02019) + (0.01241) \times (0.001325) + (0.002998) \times (0.005458) \tag{A.37}$$

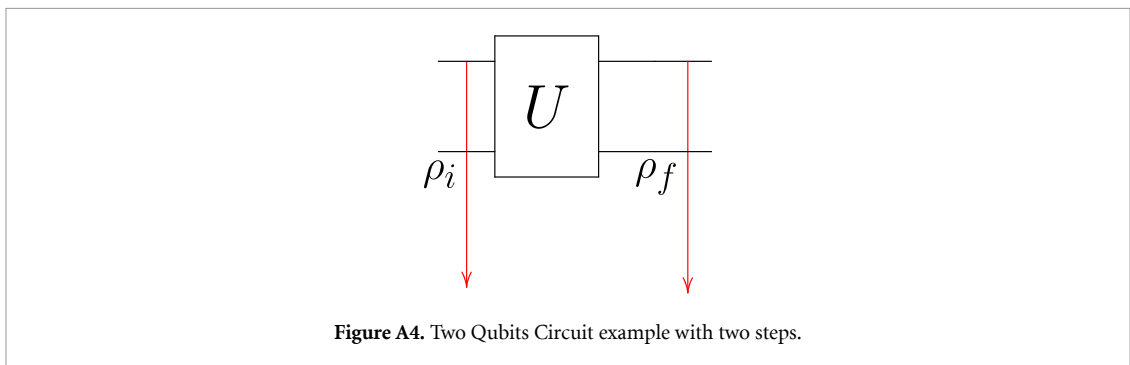
$$P_I = 0.9517. \tag{A.38}$$

As a result figure A2 becomes figure A3:

A.5. Density matrix in terms of QEC for 2 qubit

For a two-qubit quantum circuit, as shown in figure A4, the final density matrix in terms of QEC is:

$$\begin{aligned} \rho_f = & p_{ii}I \otimes IU\rho_i U^\dagger (I \otimes I)^\dagger + p_{xx}X \otimes XU\rho_i U^\dagger (X \otimes X)^\dagger + \\ & p_{yy}Y \otimes YU\rho_i U^\dagger (Y \otimes Y)^\dagger + p_{zz}Z \otimes ZU\rho_i U^\dagger (Z \otimes Z)^\dagger + \\ & p_{ix}I \otimes XU\rho_i U^\dagger (I \otimes X)^\dagger + p_{iy}I \otimes YU\rho_i U^\dagger (I \otimes Y)^\dagger + \\ & p_{iz}Z \otimes ZU\rho_i U^\dagger (I \otimes Z)^\dagger + p_{xi}X \otimes IU\rho_i U^\dagger (X \otimes I)^\dagger + \\ & p_{xy}X \otimes YU\rho_i U^\dagger (X \otimes Y)^\dagger + p_{xz}X \otimes ZU\rho_i U^\dagger (X \otimes Z)^\dagger + \\ & p_{yi}Y \otimes IU\rho_i U^\dagger (Y \otimes I)^\dagger + p_{yx}Y \otimes XU\rho_i U^\dagger (Y \otimes X)^\dagger + \\ & p_{yz}Y \otimes ZU\rho_i U^\dagger (Y \otimes Z)^\dagger + p_{zi}Z \otimes IU\rho_i U^\dagger (Z \otimes I)^\dagger + \\ & p_{zx}Z \otimes XU\rho_i U^\dagger (Z \otimes X)^\dagger + p_{zy}Z \otimes YU\rho_i U^\dagger (Z \otimes Y)^\dagger \end{aligned} \tag{A.39}$$



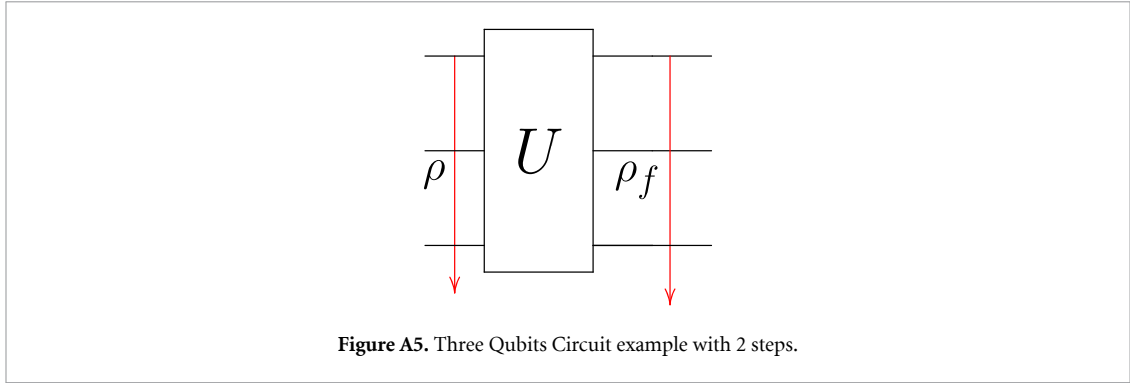
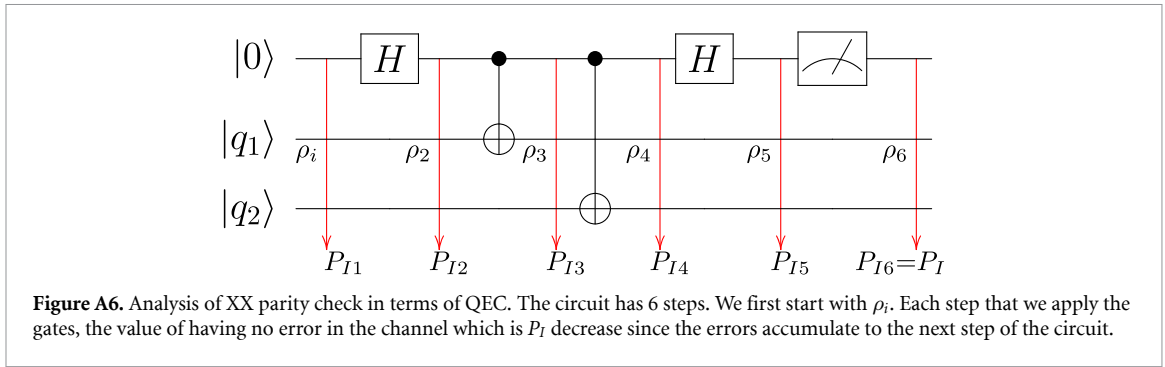


Figure A5. Three Qubits Circuit example with 2 steps.

A.6. Density matrix in terms of QEC for 3 qubit

For a three-qubit quantum circuit, as shown in figure A5, the final density matrix in terms of QEC is:

$$\begin{aligned}
 \rho_f = & p_{iii}I \otimes I \otimes IU\rho U^\dagger (I \otimes I \otimes I)^\dagger + p_{iix}I \otimes I \otimes XU\rho U^\dagger (I \otimes I \otimes X)^\dagger + \\
 & p_{iiy}I \otimes I \otimes YU\rho U^\dagger (I \otimes I \otimes Y)^\dagger + p_{iiz}I \otimes I \otimes ZU\rho U^\dagger (I \otimes I \otimes Z)^\dagger + \\
 & p_{ixi}I \otimes X \otimes IU\rho U^\dagger (I \otimes X \otimes I)^\dagger + p_{ixx}I \otimes X \otimes XU\rho U^\dagger (I \otimes X \otimes X)^\dagger + \\
 & p_{ixy}I \otimes X \otimes YU\rho U^\dagger (I \otimes X \otimes Y)^\dagger + p_{ixz}I \otimes X \otimes ZU\rho U^\dagger (I \otimes X \otimes Z)^\dagger + \\
 & p_{iyi}I \otimes Y \otimes IU\rho U^\dagger (I \otimes Y \otimes I)^\dagger + p_{iyx}I \otimes Y \otimes XU\rho U^\dagger (I \otimes Y \otimes X)^\dagger + \\
 & p_{iyy}I \otimes Y \otimes YU\rho U^\dagger (I \otimes Y \otimes Y)^\dagger + p_{iyz}I \otimes Y \otimes ZU\rho U^\dagger (I \otimes Y \otimes Z)^\dagger + \\
 & p_{izi}I \otimes Z \otimes IU\rho U^\dagger (I \otimes Z \otimes I)^\dagger + p_{izx}I \otimes Z \otimes XU\rho U^\dagger (I \otimes Z \otimes X)^\dagger + \\
 & p_{izy}I \otimes Z \otimes YU\rho U^\dagger (I \otimes Z \otimes Y)^\dagger + p_{izz}I \otimes Z \otimes ZU\rho U^\dagger (I \otimes Z \otimes Z)^\dagger + \\
 & p_{xii}X \otimes I \otimes IU\rho U^\dagger (X \otimes I \otimes I)^\dagger + p_{xix}X \otimes I \otimes XU\rho U^\dagger (X \otimes I \otimes X)^\dagger + \\
 & p_{xiy}X \otimes I \otimes YU\rho U^\dagger (X \otimes I \otimes Y)^\dagger + p_{xiz}X \otimes I \otimes ZU\rho U^\dagger (X \otimes I \otimes Z)^\dagger + \\
 & p_{xci}X \otimes X \otimes IU\rho U^\dagger (X \otimes X \otimes I)^\dagger + p_{xcx}X \otimes X \otimes XU\rho U^\dagger (X \otimes X \otimes X)^\dagger + \\
 & p_{xxy}X \otimes X \otimes YU\rho U^\dagger (X \otimes X \otimes Y)^\dagger + p_{xcz}X \otimes X \otimes ZU\rho U^\dagger (X \otimes X \otimes Z)^\dagger + \\
 & p_{xyi}X \otimes Y \otimes IU\rho U^\dagger (X \otimes Y \otimes I)^\dagger + p_{xyx}X \otimes Y \otimes XU\rho U^\dagger (X \otimes Y \otimes X)^\dagger + \\
 & p_{xyy}X \otimes Y \otimes YU\rho U^\dagger (X \otimes Y \otimes Y)^\dagger + p_{xyz}X \otimes Y \otimes ZU\rho U^\dagger (X \otimes Y \otimes Z)^\dagger + \\
 & p_{xzi}X \otimes Z \otimes IU\rho U^\dagger (X \otimes Z \otimes I)^\dagger + p_{zxx}X \otimes Z \otimes XU\rho U^\dagger (X \otimes Z \otimes X)^\dagger + \\
 & p_{xzy}X \otimes Z \otimes YU\rho U^\dagger (X \otimes Z \otimes Y)^\dagger + p_{zxx}X \otimes Z \otimes ZU\rho U^\dagger (X \otimes Z \otimes Z)^\dagger + \\
 & p_{yii}Y \otimes I \otimes IU\rho U^\dagger (Y \otimes I \otimes I)^\dagger + p_{yix}Y \otimes I \otimes XU\rho U^\dagger (Y \otimes I \otimes X)^\dagger + \\
 & p_{yiy}Y \otimes I \otimes YU\rho U^\dagger (Y \otimes I \otimes Y)^\dagger + p_{yiz}Y \otimes I \otimes ZU\rho U^\dagger (Y \otimes I \otimes Z)^\dagger + \\
 & p_{yxi}Y \otimes X \otimes IU\rho U^\dagger (Y \otimes X \otimes I)^\dagger + p_{yxx}Y \otimes X \otimes XU\rho U^\dagger (Y \otimes X \otimes X)^\dagger + \\
 & p_{yxy}Y \otimes X \otimes YU\rho U^\dagger (Y \otimes X \otimes Y)^\dagger + p_{yxx}Y \otimes X \otimes ZU\rho U^\dagger (Y \otimes X \otimes Z)^\dagger + \\
 & p_{yyi}Y \otimes Y \otimes IU\rho U^\dagger (Y \otimes Y \otimes I)^\dagger + p_{yyx}Y \otimes Y \otimes XU\rho U^\dagger (Y \otimes Y \otimes X)^\dagger + \\
 & p_{yyy}Y \otimes Y \otimes YU\rho U^\dagger (Y \otimes Y \otimes Y)^\dagger + p_{yyz}Y \otimes Y \otimes ZU\rho U^\dagger (Y \otimes Y \otimes Z)^\dagger + \\
 & p_{yzi}Y \otimes Z \otimes IU\rho U^\dagger (Y \otimes Z \otimes I)^\dagger + p_{yzx}Y \otimes Z \otimes XU\rho U^\dagger (Y \otimes Z \otimes X)^\dagger + \\
 & p_{yzy}Y \otimes Z \otimes YU\rho U^\dagger (Y \otimes Z \otimes Y)^\dagger + p_{yzz}Y \otimes Z \otimes ZU\rho U^\dagger (Y \otimes Z \otimes Z)^\dagger + \\
 & p_{zii}Z \otimes I \otimes IU\rho U^\dagger (Z \otimes I \otimes I)^\dagger + p_{zix}Z \otimes I \otimes XU\rho U^\dagger (Z \otimes I \otimes X)^\dagger + \\
 & p_{ziz}Z \otimes I \otimes YU\rho U^\dagger (Z \otimes I \otimes Y)^\dagger + p_{ziz}Z \otimes I \otimes ZU\rho U^\dagger (Z \otimes I \otimes Z)^\dagger + \\
 & p_{zxi}Z \otimes X \otimes IU\rho U^\dagger (Z \otimes X \otimes I)^\dagger + p_{zxx}Z \otimes X \otimes XU\rho U^\dagger (Z \otimes X \otimes X)^\dagger + \\
 & p_{zxy}Z \otimes X \otimes YU\rho U^\dagger (Z \otimes X \otimes Y)^\dagger + p_{zxx}Z \otimes X \otimes ZU\rho U^\dagger (Z \otimes X \otimes Z)^\dagger + \\
 & p_{zyi}Z \otimes Y \otimes IU\rho U^\dagger (Z \otimes Y \otimes I)^\dagger + p_{zyx}Z \otimes Y \otimes XU\rho U^\dagger (Z \otimes Y \otimes X)^\dagger + \\
 & p_{zyy}Z \otimes Y \otimes YU\rho U^\dagger (Z \otimes Y \otimes Y)^\dagger + p_{zyz}Z \otimes Y \otimes ZU\rho U^\dagger (Z \otimes Y \otimes Z)^\dagger + \\
 & p_{zzi}Z \otimes Z \otimes IU\rho U^\dagger (Z \otimes Z \otimes I)^\dagger + p_{zxx}Z \otimes Z \otimes XU\rho U^\dagger (Z \otimes Z \otimes X)^\dagger + \\
 & p_{zzy}Z \otimes Z \otimes YU\rho U^\dagger (Z \otimes Z \otimes Y)^\dagger + p_{zzz}Z \otimes Z \otimes ZU\rho U^\dagger (Z \otimes Z \otimes Z)^\dagger +
 \end{aligned} \tag{A.40}$$



A.7. Coefficients for dominated term from 64⁶

For the quantum circuit shown in figure A6, the total perfection rate of the circuit, P_I , is calculated by equation A.41

$$\begin{aligned}
 P_I = & p_{5iii}p_{4iii}p_{3iii}p_{2iii}p_{1iii} + p_{5iix}p_{4iix}p_{3iix}p_{2iix}p_{iix} + \\
 & p_{5iyy}p_{4iyy}p_{3iyy}p_{2iyy}p_{iyy} + p_{5iiz}p_{4iiz}p_{3iiz}p_{2iiz}p_{iiz} + \\
 & p_{5ixi}p_{4ixi}p_{3ixi}p_{2ixi}p_{ixi} + p_{5ixx}p_{4ixx}p_{3ixx}p_{2ixx}p_{ixx} + \\
 & p_{5ixy}p_{4ixy}p_{3ixy}p_{2ixy}p_{ixy} + p_{5ixz}p_{4ixz}p_{3ixz}p_{2ixz}p_{ixz} + \\
 & p_{5iyi}p_{4iyi}p_{3iyi}p_{2iyi}p_{iyi} + p_{5iyx}p_{4iyx}p_{3iyx}p_{2iyx}p_{iyx} + \\
 & p_{5iyy}p_{4iyy}p_{3iyy}p_{2iyy}p_{iyy} + p_{5iyz}p_{4iyz}p_{3iyz}p_{2iyz}p_{iyz} + \\
 & p_{5iyz}p_{4iyz}p_{3iyz}p_{2iyz}p_{iyz} + p_{5izi}p_{4izi}p_{3izi}p_{2izi}p_{izi} + \\
 & p_{5izx}p_{4izx}p_{3izx}p_{2izx}p_{izx} + p_{5izy}p_{4izy}p_{3izy}p_{2izy}p_{izy} + \\
 & p_{5izy}p_{4izy}p_{3izy}p_{2izy}p_{izy} + p_{5izz}p_{4izz}p_{3izz}p_{2izz}p_{izz} + \\
 & p_{5xii}p_{4xii}p_{3xii}p_{2xii}p_{xii} + p_{5xix}p_{4xix}p_{3xix}p_{2xix}p_{xix} + \\
 & p_{5xiy}p_{4xiy}p_{3xiy}p_{2xiy}p_{xiy} + p_{5xiz}p_{4xiz}p_{3xiz}p_{2xiz}p_{xiz} + \\
 & p_{5xxi}p_{4xxi}p_{3xxi}p_{2xxi}p_{xxi} + p_{5xxx}p_{4xxx}p_{3xxx}p_{2xxx}p_{xxx} + \\
 & p_{5xxy}p_{4xxy}p_{3xxy}p_{2xxy}p_{xxy} + p_{5xxz}p_{4xxz}p_{3xxz}p_{2xxz}p_{xxz} + \\
 & p_{5xxz}p_{4xxz}p_{3xxz}p_{2xxz}p_{xxz} + p_{5xyi}p_{4xyi}p_{3xyi}p_{2xyi}p_{xyi} + \\
 & p_{5xyx}p_{4xyx}p_{3xyx}p_{2xyx}p_{xyx} + p_{5xzi}p_{4xzi}p_{3xzi}p_{2xzi}p_{xzi} + \\
 & p_{5xzx}p_{4xzx}p_{3xzx}p_{2xzx}p_{xzx} + p_{5xzy}p_{4xzy}p_{3xzy}p_{2xzy}p_{xzy} + \\
 & p_{5xzz}p_{4xzz}p_{3xzz}p_{2xzz}p_{xzz} + p_{5yii}p_{4yii}p_{3yii}p_{2yii}p_{yii} + \\
 & p_{5yix}p_{4yix}p_{3yix}p_{2yix}p_{yix} + p_{5yiy}p_{4yiy}p_{3yiy}p_{2yiy}p_{yiy} + \\
 & p_{5yiz}p_{4yiz}p_{3yiz}p_{2yiz}p_{yiz} + p_{5yxi}p_{4yxi}p_{3yxi}p_{2yxi}p_{yxi} + \\
 & p_{5yxx}p_{4yxx}p_{3yxx}p_{2yxx}p_{yxx} + p_{5yxy}p_{4yxy}p_{3yxy}p_{2yxy}p_{yxy} + \\
 & p_{5yxx}p_{4yxx}p_{3yxx}p_{2yxx}p_{yxx} + p_{5yyi}p_{4yyi}p_{3yyi}p_{2yyi}p_{yyi} + \\
 & p_{5yyx}p_{4yyx}p_{3yyx}p_{2yyx}p_{yyx} + p_{5yyy}p_{4yyy}p_{3yyy}p_{2yyy}p_{yyy} + \\
 & p_{5yyz}p_{4yyz}p_{3yyz}p_{2yyz}p_{yyz} + p_{5yzi}p_{4yzi}p_{3yzi}p_{2yzi}p_{yzi} + \\
 & p_{5yzx}p_{4yzx}p_{3yzx}p_{2yzx}p_{yzx} + p_{5yzz}p_{4yzz}p_{3yzz}p_{2yzz}p_{yzz} + \\
 & p_{5zii}p_{4zii}p_{3zii}p_{2zii}p_{zii} + p_{5zix}p_{4zix}p_{3zix}p_{2zix}p_{zix} + \\
 & p_{5ziy}p_{4ziy}p_{3ziy}p_{2ziy}p_{ziy} + p_{5ziz}p_{4ziz}p_{3ziz}p_{2ziz}p_{ziz} + \\
 & p_{5zxi}p_{4zxi}p_{3zxi}p_{2zxi}p_{zxi} + p_{5zxx}p_{4zxx}p_{3zxx}p_{2zxx}p_{zxx} + \\
 & p_{5zxy}p_{4zxy}p_{3zxy}p_{2zxy}p_{zxy} + p_{5zxx}p_{4zxx}p_{3zxx}p_{2zxx}p_{zxx} + \\
 & p_{5zyi}p_{4zyi}p_{3zyi}p_{2zyi}p_{zyi} + p_{5zyx}p_{4zyx}p_{3zyx}p_{2zyx}p_{zyx} + \\
 & p_{5zyy}p_{4zyy}p_{3zyy}p_{2zyy}p_{zyy} + p_{5zyz}p_{4zyz}p_{3zyz}p_{2zyz}p_{zyz} + \\
 & p_{5zzi}p_{4zzi}p_{3zzi}p_{2zzi}p_{zzi} + p_{5zzx}p_{4zzx}p_{3zzx}p_{2zzx}p_{zzx} + \\
 & p_{5zzy}p_{4zzy}p_{3zzy}p_{2zzy}p_{zzy} + p_{5zzz}p_{4zzz}p_{3zzz}p_{2zzz}p_{zzz} +
 \end{aligned}
 \tag{A.41}$$

ORCID iDs

Gözde Üstün  <https://orcid.org/0009-0004-7952-636X>

Andrea Morello  <https://orcid.org/0000-0001-7445-699X>

Simon Devitt  <https://orcid.org/0000-0002-5901-1391>

References

- [1] Van Meter R and Devitt S J 2016 The path to scalable distributed quantum computing *Computer* **49** 31–42
- [2] Shor P W 1997 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM J. Comput.* **26** 1484–509
- [3] Lanyon B P et al 2010 Towards quantum chemistry on a quantum computer *Nat. Chem.* **2** 106–11
- [4] Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502
- [5] Gottesman D 2009 An introduction to quantum error correction and fault-tolerant quantum computation (arXiv:0904.2557)
- [6] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 Surface codes: towards practical large-scale quantum computation *Phys. Rev. A* **86** 032324
- [7] Bravyi S B and Kitaev A Y 1998 Quantum codes on a lattice with boundary (arXiv:quant-ph/9811052)
- [8] Dennis E, Kitaev A, Landahl A and Preskill J 2002 Topological quantum memory *J. Math. Phys.* **43** 4452–505
- [9] Erhard A et al 2021 Entangling logical qubits with lattice surgery *Nature* **589** 220–4
- [10] Stephens A M 2014 Fault-tolerant thresholds for quantum error correction with the surface code *Phys. Rev. A* **89** 022321
- [11] Gidney C, Newman M, Fowler A and Broughton M 2021 A fault-tolerant honeycomb memory *Quantum* **5** 605
- [12] Morello A, Pla J, Bertet P and Jamieson D 2020 Donor spins in silicon for quantum technologies *Adv. Quantum Technol.* **3** 2000005
- [13] Madzik M T et al 2022 Precision tomography of a three-qubit donor quantum processor in silicon *Nature* **601** 348–53
- [14] Devitt S J, Munro W J and Nemoto K 2013 Quantum error correction for beginners *Rep. Prog. Phys.* **76** 076001
- [15] Hastings M B and Haah J 2021 Dynamically generated logical qubits *Quantum* **5** 564
- [16] Bombin H 2010 Topological subsystem codes *Phys. Rev. A* **81** 032301
- [17] Kitaev A 2006 Anyons in an exactly solved model and beyond *Ann. Phys., NY* **321** 2–111
- [18] Wang D S, Fowler A G and Hollenberg L C L 2011 Surface code quantum computing with error rates over 1% *Phys. Rev. A* **83** 020302(R)
- [19] Bombin H 2010 Topological order with a twist: ising anyons from an Abelian model *Phys. Rev. Lett.* **105** 030403
- [20] Gidney C 2022 A pair measurement surface code on pentagons (arXiv:2206.12780)
- [21] Greenbaum D 2015 Introduction to quantum gate set tomography (arXiv:1509.02921)
- [22] Nielsen E, Gamble J K, Rudinger K, Scholten T, Young K and Blume-Kohout R 2021 Gate set tomography *Quantum* **5** 557
- [23] Waldherr G et al 2014 Quantum error correction in a solid-state hybrid spin register *Nature* **506** 204–7
- [24] Hensen B et al 2020 A silicon quantum-dot-coupled nuclear spin qubit *Nat. Nanotechnol.* **15** 13–17
- [25] Morello A et al 2010 Single-shot readout of an electron spin in silicon *Nature* **467** 687–91
- [26] Khaneja N, Reiss T, Kehlet C, Schulte-Herbrüggen T and Glaser S J 2005 Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms *J. Magn. Reson.* **172** 296–305
- [27] Li B, Ahmed S, Saraogi S, Lambert N, Nori F, Pitchford A and Shammah N 2022 Pulse-level noisy quantum circuits with QuTiP *Quantum* **6** 630
- [28] Ryan C A, Negrevergne C, Laforest M, Knill E and Laflamme R 2008 Liquid-state nuclear magnetic resonance as a testbed for developing quantum control methods *Phys. Rev. A* **78** 012328
- [29] Koch C P et al 2022 Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe *EPJ Quantum Technol.* **9** 19
- [30] Cory D G, Price M D, Maas W, Knill E, Laflamme R, Zurek W H, Havel T F and Somaroo S S 1998 Experimental quantum error correction *Phys. Rev. Lett.* **81** 2152–5
- [31] Zhang J, Gangloff D, Moussa O and Laflamme R 2011 Experimental quantum error correction with high fidelity *Phys. Rev. A* **84** 034303
- [32] Byrd R H, Lu P, Nocedal J and Zhu C 1995 A limited memory algorithm for bound constrained optimization *SIAM J. Sci. Comput.* **16** 1190–208
- [33] Üstün G, Morello A, Devitt S, Pitchford A 2023 Implementing-XX-single-step-parity-check-gate-on-silicon-spin-system (Version 1.0.0.) (available at: <https://github.com/apassenger/Implementing-XX-Single-Step-Parity-Check-Gate-on-Silicon-Spin-System>)
- [34] Üstün G, Morello A, Devitt S, Pitchford A 2023 CNOT gate implementation on silicon spin system using GRAPE algorithm (Version 1.0.0.) (available at: https://github.com/apassenger/CNOT_gate_implementation_with_Grape)
- [35] Üstün G, Morello A and Devitt S 2024 Single-Step-Parity-Check-Gate-Set (Version 1.0.0) Computer software (<https://doi.org/10.5281/zenodo.1234>)
- [36] Wood C J, Biamonte J D and Cory D G 2011 Tensor networks and graphical calculus for open quantum systems (arXiv:1111.6950)
- [37] Chow J M et al 2012 Universal quantum gate set approaching fault-tolerant thresholds with superconducting qubits *Phys. Rev. Lett.* **109** 060501
- [38] Korotkov A N 2013 Error matrices in quantum process tomography (arXiv:1309.6405 [quant-ph])
- [39] Cai Z and Benjamin S C 2019 Constructing smaller pauli twirling sets for arbitrary error channels *Sci. Rep.* **9** 11281
- [40] Magesan E 2008 Gaining information about a quantum channel via twirling (available at: <http://hdl.handle.net/10012/3828>)
- [41] Anwar M S, Xiao L, Short A J, Jones J A, Blazina D, Duckett S B and Carteret H A 2005 Practical implementations of twirl operations *Phys. Rev. A* **71** 032327
- [42] Hou S-Y, Sheng Y-B, Feng G-R and Long L 2014 Experimental optimal single qubit purification in an NMR quantum information processor *Sci. Rep.* **4** 6857
- [43] Behera B K, Seth S, Das A and Panigrahi P K 2019 Demonstration of entanglement purification and swapping protocol to design quantum repeater in IBM quantum computer *Quantum Inf. Process.* **18** 108
- [44] Sanders Y R, Wallman J J and Sanders B C 2015 Bounding quantum gate error rate based on reported average fidelity *New J. Phys.* **18** 012002
- [45] Qiskit contributors 2023 Qiskit: an open-source framework for quantum computing Zenodo (available at: <https://doi.org/10.5281/zenodo.2573505>)

- [46] Johansson J R, Nation P D and Nori F 2012 QuTiP: an open-source Python framework for the dynamics of open quantum systems *Comput. Phys. Commun.* **183** 1760–72
- [47] Üstün G, Morello A and Devitt S 2023 Error analysis in the context of quantum error correction (Version 1.0.0) (available at: <https://github.com/apassenger/Error-Analysis-in-QEC>)
- [48] Savvitskiy R, Botzem T, Fuentes I F, Joecker B, Hudson F E, Itoh K M, Jakob A M, Johnson B C, Jamieson D N, Dzurak A S and Morello A 2022 An electrically-driven single-atom ‘flip-flop’ qubit (arXiv:2202.04438)