Software-based Pauli Tracking in Fault-tolerant Quantum Circuits

Alexandru Paler[1] Simon Devitt[2] Kae Nemoto[2] Ilia Polian[1]

[1]Faculty of Informatics and Mathematics
University of Passau
Innstr. 43, D-94032 Passau, Germany
{alexandru.paler|ilia.polian}@uni-passau.de

[2]National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo, Japan
{devitt|nemoto}@nii.ac.jp

***Abstract*—The realisation of large-scale quantum computing is no longer simply a hardware question. The rapid development of quantum technology has resulted in dozens of control and programming problems that should be directed towards the classical computer science and engineering community. One such problem is known as Pauli tracking. Methods for implementing quantum algorithms that are compatible with crucial error correction technology utilise extensive quantum teleportation protocols. These protocols are intrinsically probabilistic and result in correction operators that occur as byproducts of teleportation. These byproduct operators do not need to be corrected in the quantum hardware itself , but are tracked through the circuit and output results *reinterpreted*. This tracking is routinely ignored in quantum information as it is assumed that tracking algorithms will eventually be developed. In this work we help fill this gap and present an algorithm for tracking byproduct operators through a quantum computation.**

I. Introduction

Quantum computing promises exponential speed-up for a number of relevant computational problems. Building a scalable and reliable quantum computer is one of the grand challenges of modern science. While small-scale quantum computers are routinely being fabricated and operated in the laboratory [1], they can only serve as feasibility studies, and fundamental breakthroughs will be required before a truly practical quantum computer can be built. As the size of computers increases, the focus of interest shifts from their basic physical principles to structured design methodologies that will allow us to realise large-scale systems [2], [3], [4], [5].

A given technology is suited for construction of a general-purpose quantum computer if it supports a direct realisation of a *universal quantum gate set* which can implement or approximate arbitrary functions [6]. Moreover, today's quantum systems exhibit high error rates and require effective *quantum error-correcting codes* (QECC) [7]. Consequently, building a practical quantum computer requires an universal gate set which can be implemented in an error-corrected manner.

In this paper, we consider a class of quantum circuits based on an universal gate set that consists of just two types of operations: injection of specific quantum states into the circuit and the controlled-not (CNOT) operation. Using the technique of *quantum teleportation*, state injections are mapped to *rotational gates* that together with the CNOT operation provide universality [8]. The advantage of this gate set is that it can be seamlessly integrated into very advanced QECC schemes, allowing for scalable, large-scale information processing [7], [9]. However, as quantum teleportation is inherently probabilistic the direction of qubit rotations is random. This randomness can be corrected via a technique known as *Pauli tracking*. Pauli tracking operates by constructing a classical record of each teleportation result and reinterprets later results during the computation. This tracking means that we do not need to perform active quantum corrections because of the probabilistic nature of teleportation operations. This technique is well known in the quantum information community and routinely ignored (referred to as working in the *Pauli frame*). While experimental results in Nuclear Magnetic Resonance systems have considered the problem of phase tracking [10], Pauli tracking is distinct as it arises from active measurement and feedforward. To our knowledge, no details on a general algorithm necessary to perform this tracking have been previously presented.

The contributions of this paper are the presentation of teleportation-based quantum computing in a generic way accesible to the design community, and the introduction of a new algorithm for *Pauli tracking*. We formalise the algorithm, prove its correctness and show its efficiency by simulations.

II. Quantum Computing

Quantum circuits represent and manipulate information in *qubits* (quantum bits). A single qubit has an associated *quantum state* $|\psi\rangle = (\alpha_0, \alpha_1)^T = \alpha_0|0\rangle + \alpha_1|1\rangle$. Here, $|0\rangle = (1,0)^T$ and $|1\rangle = (0,1)^T$ are quantum analogons of classical logic values 0 and 1, respectively. α_0 and α_1 are complex numbers called *amplitudes* with $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

A state may be modified by applying single-qubit *quantum gates*. Each quantum gate corresponds to a complex unitary matrix, and gate function is given by multiplying that matrix with the quantum state.

The application of X gate to a state results in a *bit flip*: $X(\alpha_0, \alpha_1)^T = (\alpha_1, \alpha_0)^T$. The application of the Z gate results in a *phase flip*: $Z(\alpha_0, \alpha_1)^T = (\alpha_0, -\alpha_1)^T$. Bit and phase flips are used for modelling the effects of errors on the quantum state as qubit errors can be decomposed into combination of bit and/or phase flips. Further important single-qubit quantum gates, in the context of a universal, fully error-corrected system, are $H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$, $P = \begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$ and $T = \begin{pmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{pmatrix}$. Note that $T^2 = P$ and $P^2 = Z$.

978-3-9815370-2-4/DATE14/©2014 EDAA

Quantum measurement is defined with respect to a basis and yields one of the basis vectors with a probability related to the amplitudes of the quantum state. Of importance in this work are Z- and X-measurements. Z-measurement is defined with respect to basis $(|0\rangle, |1\rangle)$. Applying a Z-measurement to a qubit in state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ yields $|0\rangle$ with probability $|\alpha_0|^2$ and $|1\rangle$ with probability $|\alpha_1|^2$. Moreover, the state $|\psi\rangle$ *collapses* into the measured state. X-measurement is defined with respect to the basis $(|+\rangle, |-\rangle)$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

A n-qubit circuit processes states represented by 2^n amplitudes, α_y, with $y \in \{0,1\}^n$ and $\sum_y |\alpha_y|^2 = 1$. Measuring multiple qubits of a circuit results in one basis vector with the probability given by the corresponding amplitude, $|\alpha_y|^2$. Quantum gates may act on several qubits simultaneously. A gate that acts on n qubits is represented by a $2^n \times 2^n$ complex unitary matrix. One important two-qubit gate is the *controlled-not* CNOT(c,t) gate, where the c qubit conditionally flips the state of the t qubits when set to $|1\rangle$.

III. Fault-Tolerant Gate Set

While every unitary complex matrix qualifies as a quantum gate in the mathematical formalism, most implementation technologies only allow a direct physical realisation of relatively few gates [11], [12], [13], [14]. Therefore, *universal gate sets* are of interest. This concept is similar to universal gate libraries in digital circuit design, where each Boolean function can be mapped to a circuit composed of, for instance, AND2 gates and inverters. One instance of a universal quantum gate set is $\{CNOT, H, P, T\}$.

A key requirement for successful realisation of quantum circuits is the ability to perform *error correction* during computation [7]. States of actual quantum systems are inherently fragile and are affected even by the slightest interaction with their environment. Therefore, QECC introduces substantial redundancy to compensate for impact on the quantum state.

A *fault-tolerant gate* for a given QECC acts directly on encoded qubits and produces legal encodings with respect to that QECC at its outputs. In self-checking design of classical circuits, error-correcting codes with this property are called *closed* with respect to the gate's operation. The closure property is advantageous because decoding and re-encoding of codewords before and after operation are avoided. As a consequence, a practical universal gate set should consist of fault-tolerant gates that allow circuit operation with errors continuously taking place.

IV. Teleporation-based Quantum Computing

In this paper, we focus on a set of operations that can be implemented in a fault-tolerant manner with respect to several state-of-the-art QECC [8]. This set consists of the CNOT gate and two *state injection* operations. State injection refers to initializing a qubit in one of the two following states $|A\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$, $|Y\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$. Using these states and a technique called *quantum teleportation*, it is possible to obtain the following three quantum gates:

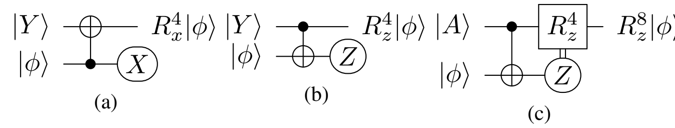

Fig. 1: Teleportation circuits used for (a) R_x^4, (b) R_z^4, (c) R_z^8

$R_x(\pi/4) = \frac{1}{\sqrt{2}}(I + iX)$, $R_z(\pi/4) = P$ and $R_z(\pi/8) = T$. In the *Bloch sphere representation* of a quantum state, $R_x(\theta)$ and $R_z(\theta)$ stand for a rotation around the X- and the Z-axis by angle θ [6]. We use the following abbreviations for brevity: $R_x^4 := R_x(\pi/4); R_z^4 := R_z(\pi/4) \equiv P; R_z^8 := R_z(\pi/8) \equiv T$. Using the relationship $H = R_z^4 R_x^4 R_z^4$, the complete universal gate set $\{CNOT, H, P, T\}$ can be obtained based on CNOT, state injection and quantum teleportation. All these gates are compatible with error-corrected, fault-tolerant computation [8], [15]. However, quantum teleportation is probabilistic itself and may require (classical) correction that will be tracked. This is described in detail below.

The rotational gates R_x^4, R_z^4 and R_z^8 are constructed by combining state injection with quantum teleportation. Applying the three employed rotational gates to an arbitrary state, $|\phi\rangle$, by quantum teleportation is shown in Fig. 1. An auxilliary qubit is initialised in state $|Y\rangle$ or $|A\rangle$ (depending on the desired rotation), and a CNOT gate is applied at the qubit that holds $|\phi\rangle$ and the auxilliary qubit (the control and target qubits are denoted by $\bullet$ and $\oplus$, respectively). Finally, a measurement (either X or Z) is performed at the control output, indicated in Fig. 1 by an encircled X or Z. The effect at the target output is shown in Fig. 1

Gate R_x^4 **:** The X-measurement in circuit of Fig. 1a yields either $|+\rangle$ or $|-\rangle$. In the former case, the rotation $R_x(\pi/4)$ has been performed correctly. If the measurement result is $|-\rangle$, the applied rotation was $R_x(-\pi/4)$. This is easily compensated by performing another rotation by angle $\pi/2$, namely applying the gate $R_x(\pi/2) = X$. Consequently, quantum teleportation must be followed by executing the X gate if the measurement result is $|-\rangle$. We call this X-*correction*.

Gate R_z^4 **:** The two possible Z-measurement results in Fig. 1b are $|0\rangle$ and $|1\rangle$. The state $|0\rangle$ indicates a correct teleportation. A measured state $|1\rangle$ is an indicator for the state $|\psi_f\rangle = \alpha_1|0\rangle - i\alpha_0|1\rangle$ where the input state was $|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$. In order to obtain the correct state, a Z operation followed by the X operation is applied, as it is easily verified that $|\psi\rangle = XZ|\psi_f\rangle = \alpha_0|0\rangle + i\alpha_1|1\rangle = R_z^4|\phi\rangle$. This operation is called XZ correction.

Gate R_z^8 **:** This gate is implemented in two stages (see Fig. 1c). The first teleporation maps state $|A\rangle$ to an intermediate state, which is then given to the R_z^4 gate from Fig. 1b that also incorporates a teleportation. The following three measurement outcomes have to be distinguished:

1) If the first measurement results in $|0\rangle$, no correction is required.
2) If the first measurement results in $|1\rangle$, the output will be used as input for a R_z^4 correctional rotation. If the

second measurement returns $|1\rangle$, no further corrections are necessary.

3) If the first measurement returns $|1\rangle$ and the second measurement yields $|0\rangle$, then an XZ correction is required.

The computation continuously requests new qubits (for injections) and abandons the old ones (the measured qubits), such that the total number of used logical qubits is n or $n+1$ (for R_z^8) at any given time.

V. Pauli Tracking Algorithm

In this section, we demonstrate how applying teleportation output corrections can be postponed to the end of calculation without losing accuracy, by considering circuits consisting of CNOT gates and the three types of rotational gates (see Fig. 1). Measurements are still performed during quantum teleportation, however their outcomes are stored in a variable rather than used for immediate correction. For each rotational gate g_i, variable b_i holds the result of the measurement. Note that $b_i \in \{|+\rangle, |-\rangle\}$ if g_i is a R_x^4 gate, $b_i \in \{|0\rangle, |1\rangle\}$ if g_i is a R_z^4 gate, $b_i \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ if g_i is a R_z^8 gate, where pairs of values refer to the outcomes of two consecutive measurements.

We derive an algorithm (see Algorithm 1) that calculates, for a given combination of b_i values, the vector of *equivalent output correction statuses* $S = (s_1, \ldots, s_n)$. For qubit k, s_k assumes one of four values that indicate the required corrections: I (no correction), X (X-correction, i.e., a bit flip), Z (Z-correction, or a phase flip), and XZ (both X- and Z-correction). The values in S are calculated such that running a teleportation-based quantum computation *without applying corrections after the gates* and applying the correction in S to the obtained output state is equivalent to teleportation-based quantum computing with immediate correction.

S is calculated by propagating (*tracking*) the correction status $(s_1, \ldots, s_n)$ through the circuit. We introduce the *correction status tracking function* τ that formalises the propagation.

There are two versions of τ: one for CNOT gates and for rotational gates. CNOT gates do not employ teleportation and therefore require no corrections; however, corrections that originated from rotational gates may show up at the inputs of a CNOT gate and have to be propagated to its outputs. Let c and t be the control and the target qubit of the CNOT gate, and let s_c^{in} and s_t^{in} be the correction statuses at the inputs of these qubits, respectively. Then, $\tau(s_c^{\text{in}}, s_t^{\text{in}})$ produces a pair of correction statuses $(s_c^{\text{out}}, s_t^{\text{out}})$ at the outputs of the CNOT gates by the following calculation:

$$s_c^{\text{out}} = \begin{cases} s_c^{\text{in}} & \text{if} \quad s_t^{\text{in}} \in \{I, X\} \\ s_c^{\text{in}} \oplus Z & \text{if} \quad s_t^{\text{in}} \in \{Z, XZ\} \end{cases} \tag{1}$$

$$s_t^{\text{out}} = \begin{cases} s_t^{\text{in}} & \text{if} \quad s_c^{\text{in}} \in \{I, Z\} \\ s_t^{\text{in}} \oplus X & \text{if} \quad s_c^{\text{in}} \in \{X, XZ\} \end{cases} \tag{2}$$

Here, $s \oplus Z$ and $s \oplus X$ are flipping the status of the respective correction in s, e.g. $XZ \oplus Z = X$, $X \oplus Z = XZ$.

For a rotational gate g_i on qubit k, the τ function takes the pre-stored measurement result b_i and the correction status s_k^{in} at its input and calculates the correction status s_k^{out} at its output. The values calculated by τ for the three types of rotational gates considered are given in Table I.

TABLE I: Correction status tracking τ for rotational gates

g_i	s_k^{in}	b_i	s_k^{out}
R_x^4	I	$\|+\rangle$	I
		$\|-\rangle$	X
	Z	$\|+\rangle$	X
		$\|-\rangle$	I
	X	$\|+\rangle$	X
		$\|-\rangle$	Z
	XZ	$\|+\rangle$	I
		$\|-\rangle$	Z
R_z^4	I	$\|0\rangle$	I
		$\|1\rangle$	XZ
	Z	$\|0\rangle$	Z
		$\|1\rangle$	X
	X	$\|0\rangle$	XZ
		$\|1\rangle$	I
	XZ	$\|0\rangle$	X
		$\|1\rangle$	X

g_i	s_k^{in}	b_i	s_k^{out}
R_z^8	I	$\|0*\rangle$	I
		$\|10\rangle$	XZ
		$\|11\rangle$	I
	Z	$\|0*\rangle$	Z
		$\|10\rangle$	X
		$\|11\rangle$	Z
	X	$\|00\rangle$	XZ
		$\|01\rangle$	I
		$\|1*\rangle$	I
	XZ	$\|00\rangle$	X
		$\|01\rangle$	Z
		$\|10\rangle$	X
		$\|11\rangle$	Z

Algorithm 1 Pauli tracking

Require: n-qubit quantum circuit with m gates $g_1, \ldots, g_m \in \{CNOT, R_x^4, R_z^4, R_x^8\}$, measurement results b_i for every rotational gate g_i

Ensure: Equivalent output correction status $S = (s_1, \ldots, s_n)$

$s_1 := s_2 := \cdots := s_n := I$;
for $i := 1$ **to** m **do**
  if g_i is a CNOT gate with control/target qubits c/t **then**
    $(s_c, s_t) := \tau(s_c, s_t)$; // Use Eqs. 1, 2
  else if g_i is a rotational gate on qubit k **then**
    $s_k := \tau(s_k, b_i)$; // Use Table I
  end if
end for
return $S = (s_1, \ldots, s_n)$;

Example: Consider the two-qubit circuit in Fig. 2. Assume that teleportation-based quantum computing has been done without applying corrections. The recorded measurement results b_i and the calculated correction statuses $S = (s_1, s_2)$ are shown in in the following table.

i		1	2	3	4	5	6
g_i		R_x^4	CNOT	R_z^8	CNOT	R_z^4	R_z^4
b_i		$\|+\rangle$	n/a	$\|10\rangle$	n/a	$\|0\rangle$	$\|1\rangle$
s_1	I	I	I	I	Z	Z	X
s_2	I	I	I	XZ	XZ	X	X

The correctness of the tracking algorithm is now formally proven. The following two lemmas formulate the validity of the τ function for the individual gates, and can be verified for every combination of inputs for τ.

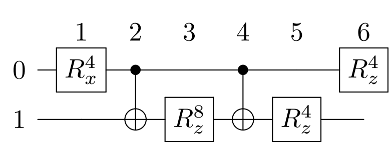

Fig. 2: Quantum circuit implemented using fault-tolerant gates

TABLE II: Run-times RT (in seconds) of the Pauli tracking algorithm for circuits with n qubits and m quantum gates

n	m	RT	n	m	RT	n	m	RT
100	1000	0	1100	1000	0.011	5100	1000	0.052
100	5000	0.002	1100	5000	0.069	5100	5000	0.309
100	10000	0.004	1100	10000	0.128	5100	10000	0.709
100	20000	0.011	1100	20000	0.300	5100	20000	1.930
100	50000	0.030	1100	50000	0.680	5100	50000	5.435

Lemma 1 Let g_i be a CNOT gate with correction status s_c^{in} at its control and s_t^{in} at its target input and $(s_c^{\text{out}}, s_t^{\text{out}}) = \tau(s_c^{\text{in}}, s_t^{\text{in}})$. Then, performing the s_c^{in}-correction at the control input and the s_t^{in}-correction at the target input followed by application of CNOT is equivalent to applying the CNOT gate first and performing s_c^{out}-correction at the control output and the s_t^{out}-correction at the target output.

Proof for $s_c^{\text{in}} = X, s_t^{\text{in}} = I$: According to Eqs. 1, 2, $(s_c^{\text{out}}, s_t^{\text{out}}) = \tau(X, I) = (X, X)$. Without loss of generality, assume that the control and target qubit of the CNOT gate are qubit 1 and 2 respectively. Then, the X-correction at the control qubit is described by matrix $X_1 = X \otimes I$ and the X-correction at the target qubit is described by matrix $X_2 = I \otimes X$. Performing the corrections first followed by the CNOT operation corresponds to the matrix $CNOT \cdot X_1 \cdot I$ while the CNOT operation followed by the two corrections are described by the matrix $X_1 \cdot X_2 \cdot CNOT$. All other combinations can be calculated similarly. $\square$

Lemma 2 Let g be a rotational gate, s^{in} the correction status at its input, b the outcome of the associated measurement and $s^{out} = \tau(s^{\text{in}}, b)$ the tracked correction status at its output according to Table I. Then, performing the s^{in} correction, applying g and performing, if needed, correction according to b, yields the equivalent state as applying g first and performing s^{out}-correction.

Proof for gate R_z^4, $b = |1\rangle$ und $s^{\text{in}} = Z$: Since $b = |1\rangle$, regular teleportation-based computing requires an XZ-correction, such that the following four operations are applied to the state: Z for s^{in}-correction; R_z^4 for gate functionality, and XZ for the correction of the wrong rotation. From Table I, $s^{\text{out}} = \tau(Z, |1\rangle) = X$ for the gate in question. The equivalence stated in the lemma is verified by showing that $XZR_z^4Z = XR_z^2$. Other cases are checked similarly. $\square$

Inductively applying the two lemmas to all gates in a circuit leads to the validity of the following theorem.

Theorem 1 Applying corrections calculated by Algorithm 1 on the state obtained without performing immediate corrections results in the same state as the state obtained when all corrections are performed immediately. $\square$

VI. Simulation Results

We implemented the Pauli tracking algorithm and applied it to a number of randomly generated quantum circuits with n logical qubits and m gates from the considered gate set. The results in Table II for $n = 100$ are indicative of largest quantum circuits within reach of today's state-of-the-art technology and $n = 1100$ and $n = 5100$ are the expected sizes of quantum computers within a decade. It can be seen that Pauli tracking is fast and all calculations can be performed within a few seconds for all cases.

The expected number of corrections without Pauli tracking is $0.5 \cdot m_4 + 0.75m_8 \approx m$, where m_4 is the number of gates R_x^4 and R_z^4 and m_8 is the number of gates R_z^8 The expected number of corrections with Pauli tracking is bounded by n, because corrections have to be performed on each output k with $s_k \neq I$. As most relevant circuits have far more gates than qubits, Pauli tracking substantially reduces the overall effort for corrections.

VII. Conclusions

We have presented an algorithm that can be used to perform Pauli tracking on quantum circuits compatible with all major classes of QECC. This result helps fill an important gap in the classical control software needed for large-scale quantum computation. Pauli tracking is instrumental for both error correction and for teleportation based protocols and this algorithm is easily adjustable to incorporate the required tracking for a specific implementation of quantum error correction. Future work will be focused on adapting this algorithm to popular error correction techniques such as Topological codes [9], [16] which requires more intensive Pauli tracking due to very high numbers of teleportation operations.

References

[1] T.D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Munro & J.L. O'Brien Quantum Computers. *Nature (London)* **464**, 45-53 (2010)

[2] Cheung, D., Maslov, D. & Severini, S. Translationt techniques between quantum circuit architectures. *Workshop on Quantum Information* (2013).

[3] Paler, A. *et al.* Synthesis of topological quantum circuits. *NANOARCH*, (2012).

[4] Saeedi, M. & Markov, I. Synthesis and optimization of reversible circuits—a survey. *ACM Computing Surveys* **45**, 21–34 (2013).

[5] Saeedi, M., Wille, R. & Drechsler, R. Synthesis of quantum circuits for linear nearest neighbor architectures. *Quantum Information Processing* **10**, 355–377 (2011).

[6] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Information*. Cambridge University Press, second edition, 2000.

[7] Lucero, S.J., Munro, W.J. & Nemoto, K. Quantum Error Correction for Beginners. *Rep. Prog. Phys* **76**, 076001 (2013).

[8] Bravyi, S. & Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A.* **71**, 022316 (2005).

[9] Fowler, A., Mariantoni, M., Martinis, J. & Cleland, A. Surface Codes, Towards practical large-scale quantum computation. *Phys. Rev. A.* **86**, 032324 (2012).

[10] Baugh, J. *et al.* Quantum Information Processing using Nuclear and Electron Magnetic Resonance: Review and Prospects. *Physics in Canada* **63**, 197-206 (2007).

[11] Devitt, S. *et al.* Architectural design for a topological cluster state quantum computer. *New. J. Phys.* **11**, 083032 (2009).

[12] Yao, N. *et al.* Scalable Architecture for a Room Temperature Solid-State Quantum Information Processor. *Nature Communications* **3**, 800 (2012).

[13] Meter, R. V., Ladd, T., Fowler, A. & Yamamoto, Y. Distributed Quantum Computation Architecture Using Semiconductor Nonophotonics. *Int. J. Quant. Inf.* **8**, 295–323 (2010).

[14] Jones, N. C. *et al.* A Layered Architecture for Quantum Computing Using Quantum Dots. *Phys. Rev. X.* **2**, 031007 (2012a).

[15] Devitt, S.J., Stephens, A.M. Munro, W.J., & Nemoto, K. Requirements for fault-tolerant factoring on an atom-optics quantum computer. *Nature (communications)* **4**, 2524, (2013).

[16] Raussendorf, R., Harrington, J. & Goyal, K. Topological fault-tolerance in cluster state quantum computation. *New J. Phys.* **9**, 199 (2007).