Quantum error correction in the noisy intermediate-scale quantum regime for sequential quantum computing

Arvid Rolander, Adam Kinos, and Andreas Walther*
Department of Physics, Lund University, SE-22100 Lund, Sweden

(Received 8 December 2021; revised 18 May 2022; accepted 24 May 2022; published 10 June 2022)

We use density-matrix simulations to study the performance of three distance three quantum error correction (QEC) codes in the context of the rare-earth (RE) ion-doped crystal platform for quantum computing. We analyze pseudothresholds for these codes when parallel operations are not available, and examine the behavior both with and without resting errors. In RE systems, resting errors can be mitigated by extending the system's ground-state coherence time. For the codes we study, we find that if the ground-state coherence time is roughly 100 times larger than the excited-state coherence time, resting errors become small enough to be negligible compared to other error sources. This leads us to the conclusion that beneficial QEC could be achieved in the RE system with the expected gate fidelities available in the noisy intermediate-scale quantum regime. However, for codes using more qubits and operations, a factor of more than 100 would be required. Furthermore, we investigate how often QEC should be performed in a circuit. We find that for early experiments in RE systems, the minimal $[\![5, 1, 3]\!]$ would be most suitable as it has a high threshold error and uses few qubits. However, when more qubits are available the $[\![9, 1, 3]\!]$ surface code might be a better option due to its higher circuit performance. Our findings are important for steering experiments to an efficient path for realizing beneficial quantum error correcting codes in early RE systems where resources are limited.

DOI: 10.1103/PhysRevA.105.062604

I. INTRODUCTION

Several platforms have been proposed for building scalable quantum computers, such as superconducting qubits and trapped ions. A promising platform, and the focus of this paper, is rare-earth (RE) ion-doped crystals [1,2]. These have some particularly attractive features, such as ground-state lifetimes of days [3] and coherence times of hours [4], high qubit connectivity [5] which enables entangling operations on non-nearest-neighbor qubits, and high spatial density, enabling efficient integration with optics. RE systems also have some unique challenges, the main one relevant to this paper being a need to perform operations in the same quantum processor node sequentially to minimize crosstalk between qubit ions. The RE system uses two hyperfine ground states as the $|0\rangle$ and $|1\rangle$ states, and a mediant excited state $|e\rangle$ is used during gate operations. The lifetime T_1 and coherence time T_2 are longer for the ground states than the excited states [3,4,6–8].

An inherent weakness of quantum computing is its sensitivity to noise, and quantum error correction (QEC) is one of the proposed solutions. Efforts have been made to study QEC in the specific context of ion traps [9–11], superconducting qubits [12,13], and systems based on nitrogen-vacancy defects in diamond [14], but it is also being discussed for ensemble qubits in stoichiometric RE-doped crystals [15]. Because even resting qubits can accumulate errors, e.g., through T_2 decoherence, some degree of parallelism of operations is seen as a requirement for beneficial QEC (see [16,17] and p. 482 in [18]). However, because the ground-state lifetime of the rare earths can be extended to several orders of magnitude longer than the time for a typical gate operation, this could potentially be relaxed in the RE case. A key point of investigation in this paper is therefore whether beneficial quantum error correction can be achieved using sequential operations, provided that the ground-state coherence time is long enough compared to the excited-state coherence time. By extending the ground-state coherence time enough, a sort of pseudoparallelism could possibly be achieved, in the sense that the resting errors are small enough compared to other error sources to be neglected. We also investigate how well the different quantum error correction protocols could be expected to perform in a rare-earth system when performing an algorithm.

We have chosen to focus on QEC in the noisy intermediate-scale quantum (NISQ) regime [19]. For the RE platform this means one quantum processor node containing around 50–100 noisy qubits, which could be available in the near future [2]. In this regime, arbitrarily high fidelity cannot be achieved through concatenated codes or large surface code lattices due to lack of resources in terms of available qubits and lower gate fidelities. Instead, it becomes important to use QEC codes with high threshold errors and using few qubits. It is still a challenge for many platforms to demonstrate beneficial QEC experimentally. An important goal is therefore to investigate

*andreas.walther@fysik.lth.se

what would be required of a RE system in order to experimentally demonstrate a gain from using QEC.

To summarize, the questions we want to answer are the following.

(1) Could beneficial QEC be demonstrated in the RE system using current protocols with projected fidelities?

(2) Can an extended resting T_2 be sufficient to overcome the demands for parallelism?

(3) How much can be gained from QEC, given reasonable parameter values?

II. SIMULATED CODES

To answer the questions posed in Sec. I, we have used density-matrix simulations (see Appendix B) to investigate three distance three QEC codes (see [18] and Appendix A1) in the context of rare-earth quantum computing. Distance 3 codes have logical codewords that are separate from each other in three places, and represent the minimum distance needed to correct for any one arbitrary error. The density-matrix approach was chosen for its ability to model arbitrary quantum processes. The QEC codes we investigate are the $[\![5, 1, 3]\!]$ code [20] using the flag syndrome extraction scheme of [21], the $[\![7, 1, 3]\!]$ Steane code [18,22] using the flag syndrome extraction scheme of [23], and the $[\![9, 1, 3]\!]$ Surface-17 code as described in [24], using the lookup table decoder of [24] with three rounds of stabilizer measurements. Because projective measurement directly on the data qubits destroys the stored information, ancillary qubits have to be used for readout. An important aspect when considering QEC in the NISQ regime is the efficient use of qubits as a resource, due to the limited number of qubits available. For this reason, the flag protocols of [21,23] are very appealing, as only two ancilla qubits are required. This can be compared to the Shor-syndrome extraction method, which would need four ancillas [25] or the Steane protocol which requires at least seven ancillas for the Steane code [26,27]. The main feature which makes the surface code attractive is its high threshold error, which could be as high as 1% [28]. Surface-17 uses 17 qubits in total, of which nine are data qubits and eight ancillas are used for measurements. However, the high qubit connectivity of the RE system makes it so that we are not limited to nearest-neighbor interactions for entanglement and parity measurements [5]. In addition to this, since we propose using fully sequential operations in each quantum processor node, we could reduce the number of ancillas required to just one, using only 9+1 qubits in total for the code we from here onward call Surface-9(1). This comes at the cost of increased run time for the error correction scheme. Table I shows the qubit requirements for each of the simulated codes with different readout schemes.

TABLE I. Overview of the QEC codes that were simulated including the number of qubits used to represent the logical state plus the number of ancilla qubits.

Code	Readout scheme or decoder	No. qubits and ancillas
$[\![5, 1, 3]\!]$	Flag	5+2
Steane $[\![7, 1, 3]\!]$	Flag	7+2
Surface-9(1) $[\![9, 1, 3]\!]$ (Surface-17)	Lookup table, three measurement rounds	9+1

III. PSEUDOTHRESHOLDS

In this section we present simulation results intended to investigate the pseudothresholds [29] of our chosen QEC codes and extraction schemes. The pseudothreshold is the break-even point for a given QEC protocol, and can be seen as the point where the logical error rate is equal to the physical error rate. The simulations were performed by applying one round of imperfect error correction, and the logical error rate was calculated as the average over all starting states corresponding to the six axes of the Bloch sphere, i.e., $|0_L\rangle$, $|1_L\rangle$, $|+_L\rangle$, $|-_L\rangle$, etc. For more information see Appendix B1.

We define the two-qubit gate (TQG) error rate p_{TQG} as the main error parameter, and error rates for other operations are defined in terms of this. The single-qubit gate (SQG) error rate p_{SQG} was set to $p_{\mathrm{TQG}}/10$, which is reasonable for RE systems [30]. We choose to treat qubit initialization and readout as single-qubit operations, and therefore the initialization and readout error rates, p_{init} and p_{RO}, were set to two-thirds of the SQG rate. Two-thirds is used because we choose to always perform initialization and readout in the $\{|0\rangle, |1\rangle\}$ basis, and they are therefore unaffected by phase errors. A summary of the values used is given in Table II, together with the optical coherence time $T_{2,\mathrm{opt}}$ and typical duration times for the different operations. The values of the operations times were chosen to resemble the ones found in [30,31]. In order to reach an initialization duration shorter than the excited-state lifetime a technique based on stimulated emission to another short-lived crystal-field level can be used [32]. For a thorough description of the error models used, see Appendix A2. In the case where a resting error is included, we assume that the damping coefficient γ is $2p_{\mathrm{TQG}}$ for TQG and initialization, p_{TQG} for SQG, and $20p_{\mathrm{TQG}}$ for readout. In other words, we assume that the resting errors reduce as the two-qubit error reduces. The relative ratios between the different resting errors come directly from the ratios of their durations.

Figure 1(a) shows how the logical error rate varies as a function of p_{TQG}, i.e., the two-qubit error rate, when all error sources are included, and Fig. 1(b) shows the case when resting errors are excluded. The case with no resting errors is relevant since the ground-state coherence time of RE systems is much longer then the gate coherence time. Black, vertical lines mark the points where the physical TQG error rate is $p_{\mathrm{TQG}} = 10^{-3}$. Interestingly, the $[\![5, 1, 3]\!]$, not Surface-9(1), performs best both when resting errors are included and excluded. There are several reasons for this, but the main one seems to be the comparatively low SQG error that we have used. Lowering the SQG error compared to the TQG error improves the performance of all codes, but the effect is larger for the $[\![5, 1, 3]\!]$ code than the other two. This discrepancy can be explained by looking at what proportion of the operations used in each protocol are single-qubit operations: the $[\![5, 1, 3]\!]$ uses 60% single-qubit operations, Surface-9(1) uses 50%, and

TABLE II. Parameter values used in simulations [30,31].

Parameter	Duration t_{dur} (μs)	Error symbol	Error value	Resting error [Fig. 1(a)]	Resting error (Fig. 2)
Single-qubit gate	5	p_{SQG}	$p_{\text{TQG}}/10$	p_{TQG}	$\gamma = e^{\frac{-t_{\text{dur}}}{2T_{2,\text{spin}}}}$
Two-qubit gate	10	p_{TQG}	p_{TQG}	$2p_{\text{TQG}}$	$\gamma = e^{\frac{-t_{\text{dur}}}{2T_{2,\text{spin}}}}$
Readout	100	p_{RO}	$p_{\text{TQG}}2/30$	$20p_{\text{TQG}}$	$\gamma = e^{\frac{-t_{\text{dur}}}{2T_{2,\text{spin}}}}$
Initialization	10	p_{init}	$p_{\text{TQG}}2/30$	$2p_{\text{TQG}}$	$\gamma = e^{\frac{-t_{\text{dur}}}{2T_{2,\text{spin}}}}$

the Steane code uses between 50 and 62.5% depending on the syndrome measurement results. Another reason for the lower surface code performance could be the choice of the lookup table decoder, as other decoders could yield better results [12]. Figure 1 also shows that the surface code and Steane code are more affected by resting errors than the $[\![5, 1, 3]\!]$ code. This can be explained by looking at the number of operations and qubits used in each of the protocols; the $[\![5, 1, 3]\!]$ uses 55 operations in the best case and 100 operations in the worst case scenario, and the Steane code uses 96 operations in the best case and 276 operations in the worst case. Surface-9(1) always uses 144 operations. The different cases arise as the flag protocols follow different paths depending on if the circuits flag or not. Since all operations are performed sequentially, this means that Surface-9(1) and the Steane code have more idle time where resting errors are accumulated on more qubits compared to the $[\![5, 1, 3]\!]$ code. The higher number of qubits used by the Steane code and Surface-9(1) compared to the $[\![5, 1, 3]\!]$ code also contributes to the higher error, as they have more qubits in waiting for every operation.

IV. PSEUDOPARALLELISM

In this section we present results aiming to investigate pseudoparallelism by varying the magnitude of resting errors. The simulations were again performed by applying one round of imperfect error correction, followed by calculating the error rate as the average over the six axes of the Bloch sphere. For these simulations, we used a different model for resting errors than that of Sec. III. Instead of assuming the

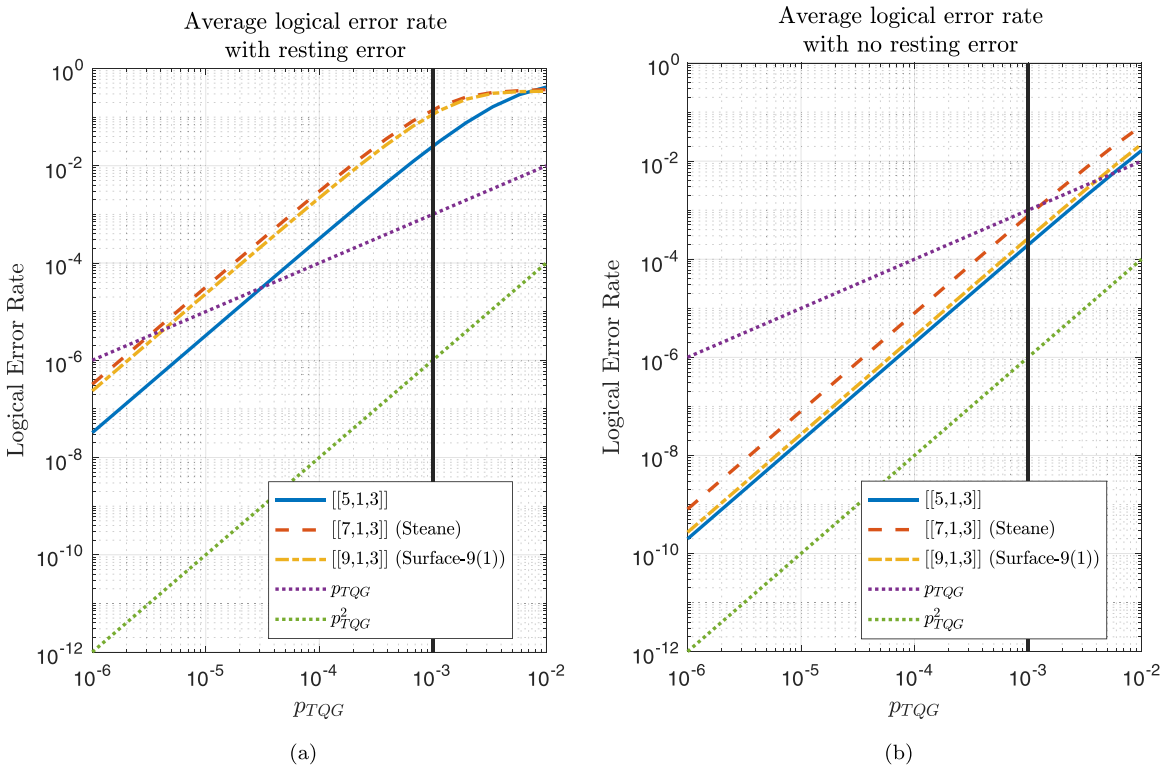

FIG. 1. Pseudothresholds for our chosen codes with resting errors either included (a) or excluded (b). The logical error rates were calculated as the average for starting states corresponding to the six axes of the Bloch sphere. In panel (a), we have assumed that the resting errors for the different operations scale with p_{TQG}. This assumes that if we are able to improve p_{TQG}, we will be able to improve the resting errors by the same factor. Black, vertical lines mark the points where the physical two-qubit error rate is $p_{\text{TQG}} = 10^{-3}$. This is approximately the error rate where a physical system, with today's protocols [2], could be expected to lie [30]. The purple (top), dotted lines show p_{TQG}, and the green (bottom) dotted lines show p_{TQG}^2.

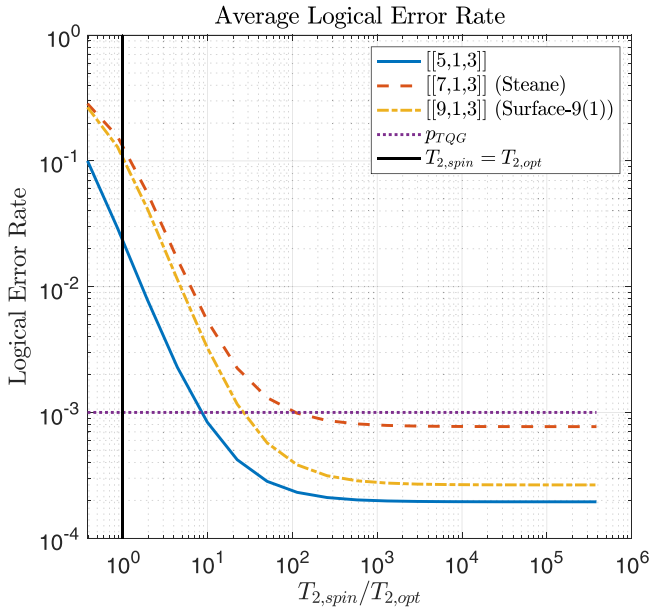

FIG. 2. Logical error rate as a function of the ground-state coherence time $T_{2,\mathrm{spin}}$ which varies between 1 ms and 1 h. The x axis is scaled by the excited-state coherence time $T_{2,\mathrm{opt}} = 2.5$ ms. In this simulation we have used a different model of the resting error compared to the one used in Fig. 1(a). Here, we assume a constant value of the two-qubit error rate, $p_{\mathrm{TQG}} = 10^{-3}$, but the resting errors still change due to changes in $T_{2,\mathrm{spin}}$ (see Table II). This model therefore assumes that it is possible to reduce resting errors even though the two-qubit error rate is kept constant. The models give the same values for all errors when $p_{\mathrm{TQG}} = 10^{-3}$ and $T_{2,\mathrm{spin}} = 2.5$ ms, which are reasonable values [30] for a RE system using current protocols [2]. These points are marked with black, vertical lines in both Figs. 1(a) and 2. Furthermore, in a RE system it is possible to reach values for $T_{2,\mathrm{spin}}$ of hours [4]. This corresponds to points to the far right in Fig. 2, where the resting errors are small enough compared to other error sources to be neglected. The errors here have approximately the same values as in Fig. 1(b) when $p_{\mathrm{TQG}} = 10^{-3}$, which is marked with a black line in Fig. 1(b). This shows that the resting errors can be reduced enough to become negligible.

resting errors follow p_{TQG}, we kept the TQG error rate fixed at $p_{\mathrm{TQG}} = 10^{-3}$, but assumed that we can improve the resting errors by increasing the ground-state coherence time $T_{2,\mathrm{spin}}$. Therefore, the magnitudes of the resting errors were calculated as $\gamma = 1 - e^{\frac{-t_{\mathrm{dur}}}{2T_{2,\mathrm{spin}}}}$ (see Appendix A2b) with the duration times found in Table II.

The simulation results are shown in Fig. 2, where we have chosen to scale the ground-state coherence time $T_{2,\mathrm{spin}}$ by the excited-state coherence time $T_{2,\mathrm{opt}} = 2.5$ ms. The point where $T_{2,\mathrm{spin}}/T_{2,\mathrm{opt}} = 1$ is marked with a black, vertical line. At this point, the error rate and magnitude of the resting errors are approximately the same as in Fig. 1(a) at the vertical, black line where $p_{\mathrm{TQG}} = 10^{-3}$. As the value of $T_{2,\mathrm{spin}}/T_{2,\mathrm{opt}}$ increases, a plateau is reached where the resting errors are insignificant, and the logical error rate is essentially the same as at the vertical black line in Fig. 1(b), where resting errors are not included. This shows that even under the constraint that all operations in a quantum processor node must be performed sequentially, resting errors can be reduced to the point of being negligible in comparison with other error sources.

In a RE system the $T_{2,\mathrm{spin}}$ can be improved by various actions, such as applying a magnetic field in the right direction [4] or applying dynamic decoupling sequences [33]. However, these actions have a cost; for instance, applying a magnetic field would also split the atomic levels, making the qubit space more complicated, and running dynamic decoupling sequences would make the gate sequences more involved. For these reasons, it is interesting to see that there is a level beyond which no further gain to the resting $T_{2,\mathrm{spin}}$ can be obtained. This allows experimentalists to hit a specific target rather than arbitrarily increasing $T_{2,\mathrm{spin}}$, which serves to minimize the experimental efforts required to realize a quantum computer.

It is important to note that such a target $T_{2,\mathrm{spin}}$ will change for higher distance codes, but we can estimate by how much: each time we perform an operation on one of the data qubits, we assume that there is a probability in the order of p_{TQG} for a gate error to occur. If we view resting errors as Z errors (see Appendix A2b) we can say that the probability for a single resting error to occur on just one of the qubits is roughly $(N-1)p_{\mathrm{rest}}$, where N is the number of qubits, p_{rest} is the probability for a resting error on a single qubit, and we assume that higher orders of p_{rest} are negligible. To be able to neglect resting errors, the probability for them occurring must be significantly smaller than the probability of a gate error. Studying Fig. 2, we see that the error rates flatten at roughly $T_{2,\mathrm{spin}} = 100T_{2,\mathrm{opt}}$. Since the codes we use here use approximately ten qubits, this means that resting errors become negligible approximately when the probability of only one resting error occurring on any of the qubits is an order of magnitude lower than the probability of a gate error, which seems reasonable.

In the NISQ regime, we aim for having roughly 100 total qubits. Thus, implementing a distance 11 Surface code might be a reasonable future goal. The distance 11 Surface code uses $11^2 = 121$ data qubits, which, by our argument, would require a $T_{2,\mathrm{spin}}$ that is roughly ten times longer than that required for Surface-9(1), i.e., $T_{2,\mathrm{spin}} = 1000T_{2,\mathrm{opt}}$. With $T_{2,\mathrm{opt}} = 2.5$ ms this means the distance 11 code would require $T_{2,\mathrm{spin}} = 2.5$ s. Achieving a $T_{2,\mathrm{spin}}$ of this magnitude is still very realistic in RE systems since coherence times of hours have been reached experimentally [4], and we therefore see reason to be optimistic. Although it can be noted that the method used to increase the $T_{2,\mathrm{spin}}$ might have consequences, such as splitting levels or complicating sequences, and it remains for future studies to determine the most optimal way to accomplish this. It is also important to note that a typical RE quantum processing node will likely have around 100 qubits [5], and operations in different nodes will be able to be performed in parallel [2]. This means that as long as we can achieve a $T_{2,\mathrm{spin}}$ that is long enough for 100 qubits, scaling up further will be possible.

Lastly, as a side note, the main result presented in this section could be valid for other systems than REs; i.e., as long as the idle qubit T_2 (spin) is sufficiently longer than the T_2 relevant during gate operations (optical) its effect on the logical gate error is negligible and a pseudoparallelism is achieved.

V. COMPARISON WITH PHYSICAL CIRCUIT

In this section we present results meant to study the performance of the error correcting codes when used in a circuit. From the results of Sec. IV we can expect that it is experimentally feasible to be able to reach a point where the ground-state coherence time is not the main limitation when performing error correction. For this reason, we will omit ground-state dephasing in the following. The simulated experiment consists of applying a logical gate (X_L if the initial state is $|0_L\rangle$ or $|1_L\rangle$, and Z_L otherwise) n_g times, which for distance 3 codes corresponds to three physical Pauli gates, followed by a round of error correction. The logical error rate is then calculated as the average for starting states corresponding to the six axes of the Bloch sphere as before. We again used the ratios for the physical error rates given in Table II.

To be able to compare the performance of the logical circuit to the equivalent physical circuit without error correction, we define a gain parameter as the ratio of the total error of the physical circuit and the total error of the logical circuit. This parameter tells us by which factor the error per gate changes when error correction is used compared to the physical circuit without error correction.

At this point, it seems reasonable to ask how the gain should behave as a function of n_g. From a more holistic perspective, it could be argued that higher values of n_g should give higher gain since error correction has considerable overhead in a number of operations. For small values of n_g the contribution of this overhead to the total error could be large in comparison to the contribution from increasing n_g. However, for large values of n_g both the logical and physical qubit will approach a completely mixed state, and the gain should equal 1 as there will be no way of guessing the correct state. Therefore, we would expect to find an optimal value.

We can put this in more mathematical terms; since all codes we study correct one error, the dominant term in the logical error rate is proportional to the probability of getting a weight 2 error, i.e., an error acting on two data qubits. We use proportional here, since some weight 2 errors are equal to a weight 1 error modulo one of the code stabilizers. Call the logical error rate ϵ_L, and the number of operations used for error correction N. Assuming that all operations have an error rate of p and neglecting higher-order terms, we then have

$$\epsilon_L \propto p^2 \binom{N}{2}. \tag{1}$$

Adding n_g logical Pauli gates before error correction, and assuming that the physical operations again have error rate p, we get

$$\epsilon_L \propto p^2 \binom{N+3n_g}{2} = \frac{p^2}{2}(N+3n_g)(N+3n_g-1), \tag{2}$$

since a logical Pauli gate on a distance 3 code corresponds to three physical Pauli gates. In the physical case, however, the total error rate ϵ after n_g operations, again neglecting higher-order terms, is

$$\epsilon \propto p \binom{n_g}{1} = p n_g. \tag{3}$$

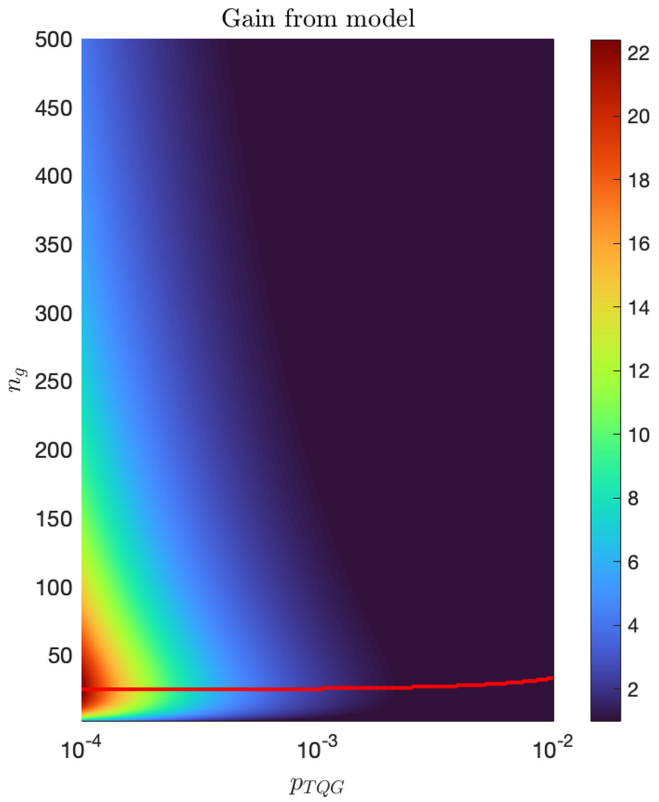

FIG. 3. Gain, defined as the physical error when not using error correction divided by the logical error when using error correction calculated from Eq. (4) with $C = 1$ and $N = 75$. Here, we have assumed that all operations have the same error rate, p_{TQG}. The number of logical gates performed before error correction, as well as the number of physical gates, is given by n_g. The red line marks the value of n_g which maximizes the gain for every given p_{TQG}. As is shown in Eq. (4), below the threshold the maximum gain is achieved for a constant value of n_g, independent of p_{TQG}.

Thus, the gain g can be written

$$g = \frac{\epsilon}{\epsilon_L} = \frac{C}{p}\frac{n_g}{(N+3n_g)(N+3n_g-1)}, \tag{4}$$

where C is a constant determined by the proportion of weight 2 errors that are not equivalent to a weight 1 error modulo a stabilizer. With the restriction that $n_g \geqslant 0$ and $N \gg 1$ for error correction, the gain function is concave and we can conclude that it has a maximum. Moreover, we can conclude that below the threshold, the gain is maximized for a constant value of n_g independent of p_{TQG}.

Figure 3 shows a color map plot of a modified version of Eq. (4), with a second-order term added to the numerator, against both n_g and p, with $N = 75$ and $C = 1$. The addition of the second-order term captures the behavior of the gain function for values of p above the threshold, where no gain from error correction can be expected and the optimal value of n_g would therefore be one which gives a completely mixed state. In this model, the gain can be seen to achieve a maximum at $n_g = 25$, i.e., $N/3$ for values of p below the threshold. This means that in this simplified model where all operations have equal error rates, the optimal value of n_g is the one where the number of physical gates used before error correction is $N/3$.

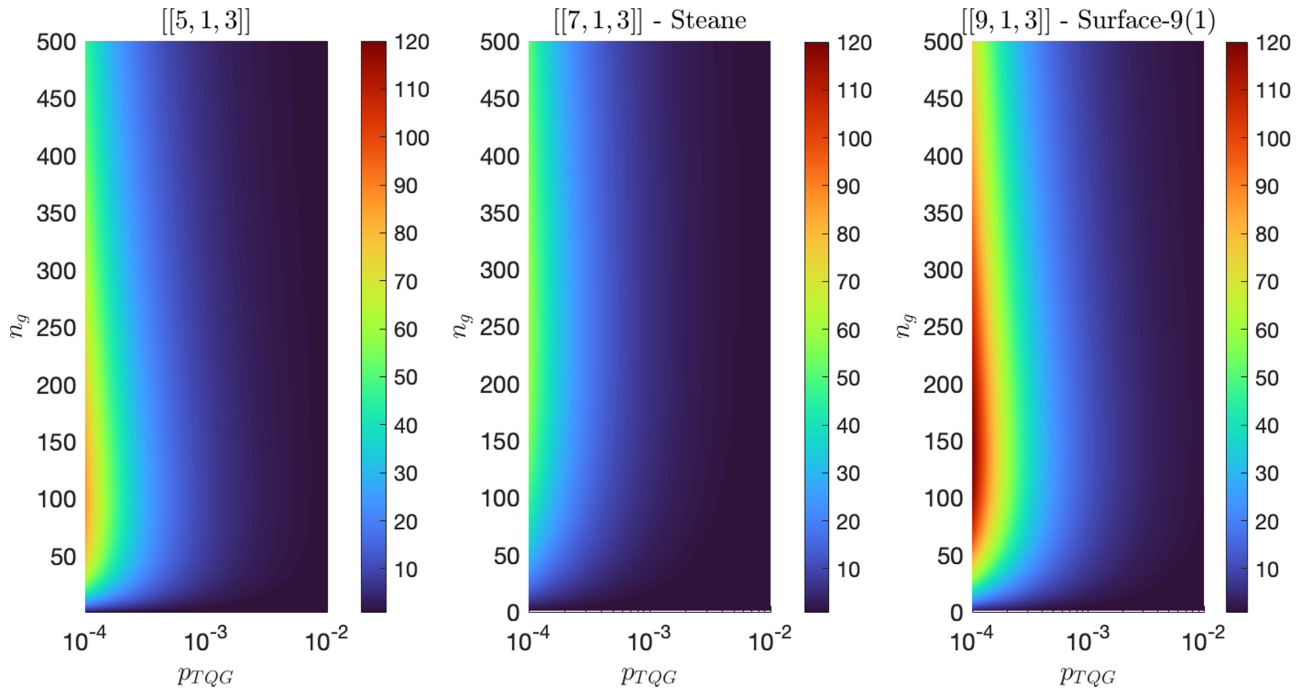

FIG. 4. Gain of the logical circuit with error correction compared to the physical circuit. The $[\![5, 1, 3]\!]$ code achieves maximum gain at $n_g = 99$, the Steane code achieves maximum gain at $n_g = 245$, and Surface-9(1) achieves maximum gain at $n_g = 142$.

This means that the optimal number of operations used before error correction is the same as the number of operations used in error correction. Moreover, the gain increases by one order of magnitude when there is a reduction in p of one order of magnitude. This agrees well with what we would expect, as the error of a distance 3 code should scale as p_{TQG}^2.

Our simple model can be compared to the full simulations of the different codes, which can be seen in Fig. 4 where we plot the gain as a color map against both the gate number n_g and the error rate p_{TQG}. The qualitative behavior is very similar to that of Fig. 3, with the gain achieving a maximum for a constant value of n_g below the threshold error rate. It should be noted that here, the optimal n_g is not just the one which corresponds to the number of operations used in the error correction protocol. This is because different types of operations, e.g., TQG and SQG, are assigned different error rates.

While Fig. 4 contains only the gain, the interested reader can see Fig. 5 in Appendix C for the corresponding logical and physical error rates using the optimal n_g for each of the three codes. However, when optimizing the error correcting codes it is mostly the gain itself that matters. A technologically relevant algorithm might require logical errors per gate less than 10^{-15} [34,35], so at our level of error the gain is important even when the errors are very small. For high error levels of around 50% the success rate of the algorithm breaks down, but even at relatively high total errors of, e.g., 10%, the entire algorithm could still be successful by repeating it a couple of times to exponentially suppress the error at a cost of only a linear overhead.

When it comes to the performance of the codes, it is clear that both the $[\![5, 1, 3]\!]$ and Surface-9(1) codes outperform the Steane code with a large margin, both in terms of the threshold error and the maximum gain. The $[\![5, 1, 3]\!]$ and Surface-9(1) codes have very similar threshold values, which are slightly higher than those found in Sec. III as discussed more in Appendix C. Furthermore, the maximum gain of Surface-9(1) is higher than that of $[\![5, 1, 3]\!]$.

With Eq. (4) in mind, we can say something about why these differences occur. We can see that the characteristics of a QEC code are determined by N, C, and also, in our case, the share of single-qubit operations of N. The Steane code in our case has a high maximum N compared to the other two, and C apparently does not compensate for this. In the case of Surface-9(1), it is clear that the value of C compensates for the higher number of operations it uses compared to the $[\![5, 1, 3]\!]$ code. This is likely also the reason why Surface-9(1) achieves a higher gain than the $[\![5, 1, 3]\!]$ code: if more weight 2 errors are correctable by the code, adding more error locations in the form of physical gates should have less of an impact.

One thing to take into account when running experiments is the fact that the gain decreases slowly for values of n_g above the optimal value. This means that the error correction overhead might be reduced by increasing n_g above the optimal value, for only a small cost in decreased gain. This depends on the SQG error, however, as the gain will decrease faster for higher SQG errors. This can be seen in Fig. 3, where all errors are the same. Assuming an experimental RE system can reach a value of $p_{\mathrm{TQG}} = 10^{-3}$, the possible gain in error per gate is approximately 20 for both the $[\![5, 1, 3]\!]$ code and Surface-9(1), and approximately 10 for the Steane code.

VI. CONCLUSIONS

We have studied the performance of three QEC codes, namely, the minimal $[\![5, 1, 3]\!]$ code, the $[\![7, 1, 3]\!]$ Steane code, and the $[\![9, 1, 3]\!]$ Surface-9(1) code, using density-matrix simulations, in the context of RE quantum computing. Since each

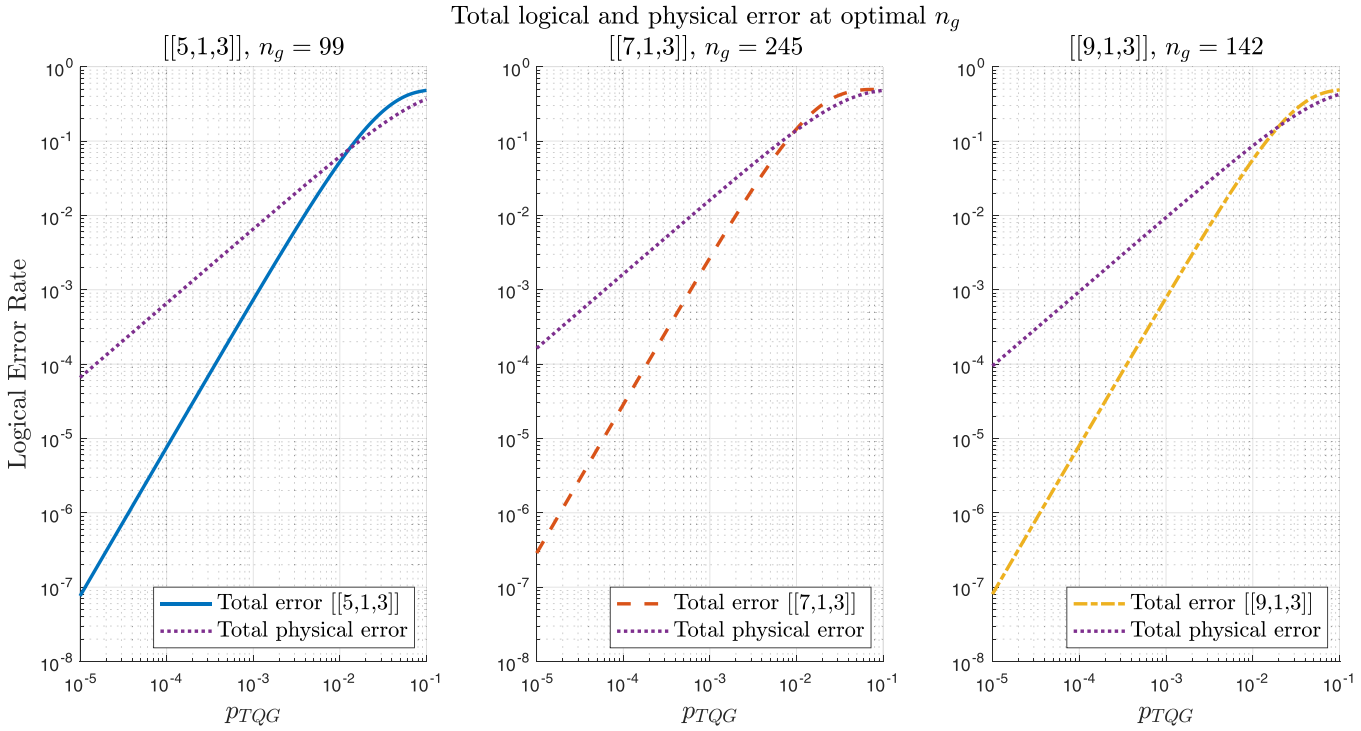

FIG. 5. Total error of the logical circuit with QEC compared to the physical circuit for the optimal n_g which maximizes the gain shown in Fig. 4. The $[\![5, 1, 3]\!]$ code uses $n_g = 99$, the Steane code uses $n_g = 245$, and Surface-9(1) uses $n_g = 142$.

local processor node in a RE quantum computer requires sequential gate operations, the values of the pseudothresholds of all codes are highly dependent on whether resting errors are included or not. It was found that with resting errors and using currently expected gate fidelities, none of the codes would allow a RE quantum computer to stay below the pseudothresholds required for fault tolerant operation. However, if the ground-state coherence time $T_{2,\text{spin}}$ is sufficiently long, the magnitude of the resting errors is reduced to a point where they are negligible in comparison to other error sources, and beneficial QEC would be possible. For the codes and protocols we studied, it is sufficient that the $T_{2,\text{spin}}$ is around two orders of magnitude larger than the $T_{2,\text{opt}}$, while a longer $T_{2,\text{spin}}$ is required for higher distance codes. Such long $T_{2,\text{spin}}$ are feasible to achieve experimentally [4].

We also compared the error per gate when simulating QEC on a physical circuit. We found that the optimal number of gates is determined by the number of operations used by the QEC protocol. The Steane code was found to be inferior to the other options both in terms of its pseudothreshold and in terms of the improvement in error rate compared to a physical circuit. The $[\![5, 1, 3]\!]$ code and Surface-9(1) were found to have very similar pseudothresholds of around $p_{\text{TQG}} = 5 \times 10^{-3}$ and 4×10^{-3}, respectively. In terms of gain compared to a physical circuit, Surface-9(1) was found to be slightly better than the $[\![5, 1, 3]\!]$ code but at $p_{\text{TQG}} = 10^{-3}$ both had gain values of around 20, compared to a gain value of around 10 for the Steane code. Given the constraints of working in the NISQ regime, an experimentalist would likely be better off demonstrating QEC with the $[\![5, 1, 3]\!]$ code rather than Surface-9(1) in an early phase. This is because the $[\![5, 1, 3]\!]$ code uses fewer qubits while still having a similar threshold error. As the number of available qubits increases, the surface codes become more attractive. This is partly due to their higher gain, but also the straightforward way in which higher distance codes can be implemented.

Finally, a large-scale RE quantum computer is envisioned to consist of many processor nodes similar to the ones we discuss in the present paper, each containing around 100 qubits [5], where operations in different nodes could run in parallel [2]. Thus, how to realize QEC on a RE quantum computer beyond the NISQ regime is beyond the scope of this paper, but it is an important question for future research.

ACKNOWLEDGMENTS

We acknowledge helpful discussions and comments from Stefan Kröll, Lars Rippe, and Klaus Mølmer. The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant No. 820391 (SQUARE), as well as from the Swedish Research Council (Grants No. 2015-03989 and No. 2019-04949).

APPENDIX A: BACKGROUND AND THEORY

1. Quantum error correction

In this section we give an overview of QEC that is relevant for this paper.

Stabilizer codes

All of the QEC codes used in this paper belong to a class of codes known as *stabilizer codes*. These can be compactly

described in terms of a number of commuting operators belonging to the Pauli group, consisting of tensor products of the Pauli operators X, Y, and Z together with the identity I and multiplicative factors ± 1 and $\pm i$ [18]. These operators are known as the *stabilizers* of the code [18,36]. Logical code words $|0_L\rangle$ and $|1_L\rangle$ are defined as simultaneous $+1$ eigenstates of all of the operators in the stabilizer [18,36,37]. Errors can be detected and characterized through parity measurements of the stabilizers, and based on the measured *error syndrome* S_i a correction operator can be applied [17]. The error syndrome consists of the recorded parities of all of the stabilizer measurements, and can be seen as a binary word, e.g., $S_i = (1 \quad 1 \quad -1) \leftrightarrow (0 \quad 0 \quad 1)$ for a hypothetical code with only three stabilizers. The error syndromes can be conveniently labeled by the corresponding decimal number, so that, e.g., $S_3 = (0 \quad 1 \quad 1)$.

Quantum error correcting codes can also be described in the form $[\![n, k, d]\!]$, where n is the number of physical qubits used for encoding, k is the number of encoded qubits, and d is called the *distance* of the code. The distance d is a measure of how different the code words of a given code are. The distance determines how many errors a code can correct, since a distance of $d = 2t + 1$ is required to correct t errors. The distance can be seen as the minimum weight of any logical operator on the code, that is, the minimum number of qubits on which a logical operator on the code differs from the identity. Logical operators for stabilizer codes can be defined by picking operators which commute with all members of the stabilizer group, and which obey the commutation relations of the equivalent physical operators. One of the most famous examples of quantum error correcting codes, the Steane code, is a $[\![7, 1, 3]\!]$ code, which means it uses seven qubits to encode one logical qubit with distance 3 [17,18,36].

2. Error models

In this section we describe the error model used throughout the simulated experiment.

a. Single- and two-qubit gate errors

We have modeled SQG errors by assuming a perfect gate is followed by an error process $\mathcal{E}_{\mathrm{SQG}}$, given by

$$\mathcal{E}_{\mathrm{SQG}}(\rho) = (1 - p_X - p_Y - p_Z)\rho + p_X X\rho X^\dagger + p_Y Y\rho Y^\dagger + p_Z Z\rho Z^\dagger, \quad \text{(A1)}$$

where I, X, Y, and Z are the Pauli operators with corresponding Pauli matrices σ_I, σ_X, σ_Y, and σ_Z. Typically, we set a value p_{SQG} for the total SQG error rate and use $p_X = p_Y = p_Z = p_{\mathrm{SQG}}/3$, i.e., the standard single-qubit depolarizing channel [18,37]. For TQGs, such as the controlled-NOT gate, we assume that the error operators are given by the nontrivial two-qubit Pauli group operators, i.e., tensor products of the form $\sigma_i \otimes \sigma_j$ with corresponding probabilities given by p_{ij}, with i, j being I, X, Y, or Z. The error process $\mathcal{E}_{\mathrm{TQG}}$ for the TQG is given by

$$\mathcal{E}_{\mathrm{TQG}}(\rho) = \left(1 - \sum_{ij\neq II} p_{ij}\right)\rho + \sum_{ij\neq II} p_{ij}\sigma_i \otimes \sigma_j \rho \sigma_i^\dagger \otimes \sigma_j^\dagger. \quad \text{(A2)}$$

For the TQG we typically set an error rate p_{TQG} and use $p_{ij\neq II} = p_{\mathrm{TQG}}/15$ corresponding to a two-qubit depolarizing channel [37]. The full operation for an arbitrary gate $\mathcal{G}$ is thus given by

$$G(\rho) = \mathcal{E}(\sigma_G \rho \sigma_G^\dagger), \quad \text{(A3)}$$

where σ_G is the operation matrix for the error-free G.

b. Idle qubit errors

Because the ground-state lifetime of the rare earths can be days, amplitude damping is neglected on resting qubits. Phase damping is considered, and since operations are assumed to be sequential, it is applied on all qubits except the involved qubits of a gate during gate operations, where instead normal gate errors are applied. Phase damping is also applied to all qubits except the target during initialization and readout.

The standard form of the phase damping channel for one qubit is given by the operation matrices [18]

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad \text{(A4)}$$

$$E_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\gamma} \end{pmatrix}. \quad \text{(A5)}$$

However, the channel can be equivalently represented by the operation matrices

$$\tilde{E}_0 = \sqrt{\alpha}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{(A6)}$$

$$\tilde{E}_1 = \sqrt{1-\alpha}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \text{(A7)}$$

where $\alpha = (1 + \sqrt{1-\gamma})/2$ [18]. We use the latter of these representations, since it can save some calculations during simulations. In the case where the ground-state coherence time $T_{2,\mathrm{spin}}$ is used to calculate the damping coefficient, we use

$$\gamma = 1 - e^{-t/2T_{2,\mathrm{spin}}}, \quad \text{(A8)}$$

where t is the idle time given by the duration of the operation. The different operations used are SQG, TQG, initialization, and readout, and we denote their operation times by t_{SQG}, t_{TQG}, t_{init}, and t_{RO}, respectively.

c. Initialization and readout errors

Qubits are for simplicity always initialized in the $|0\rangle$ state, and it is assumed that there is a probability, p_{init}, of an initialization error when it is instead initialized as $|1\rangle$. The density matrix ρ after initialization for a single qubit with initialization error rate p_{init} is thus

$$\rho = (1 - p_{\mathrm{init}})|0\rangle\langle 0| + p_{\mathrm{init}}|1\rangle\langle 1| := \begin{pmatrix} 1 - p_{\mathrm{init}} & 0 \\ 0 & p_{\mathrm{init}} \end{pmatrix}. \quad \text{(A9)}$$

We assume that readout in the computational basis $|0\rangle, |1\rangle$ is subject to a declaration error, e.g., where a measurement result of -1 is incorrectly declared as $+1$ and vice versa. In the general case we use a symmetric declaration error, where the states $\rho_\pm$ after measurement of $+1$ and -1 are

given by

$$\rho_+ = (1 - p_{\mathrm{RO}})p_+\rho_{+1} + p_{\mathrm{RO}}p_-\rho_{-1}, \tag{A10}$$

$$\rho_- = (1 - p_{\mathrm{RO}})p_-\rho_{-1} + p_{\mathrm{RO}}p_+\rho_{+1}, \tag{A11}$$

where $\rho_{\pm 1}$ are the resulting density matrices after error-free measurement, $p_\pm$ is the probability to measure $+1$ and -1, and p_{RO} is the probability of a declaration error. An idiosyncrasy of the RE system is that declaration errors are asymmetrical, so that a measurement result of -1 has a probability to be declared as $+1$, but a declaration of -1 is always correct. This is due to the nature of the readout via the dipole mechanism where only one of the states is excited [31]. The states $\rho_\pm$ after measuring $+1$ and -1 are given by

$$\rho_+ = p_+\rho_{+1} + p_{\mathrm{RO}}p_-\rho_{-1}, \tag{A12}$$

$$\rho_- = (1 - p_{\mathrm{RO}})p_-\rho_{-1} \tag{A13}$$

in the asymmetrical case.

APPENDIX B: DENSITY-MATRIX SIMULATION

The density-matrix simulations were performed using a MATLAB framework. The density-matrix approach was chosen for its ability to model arbitrary quantum processes, such as, for example, T_2 decoherence [12]. It would also be possible in future implementations to find exact representations for all gate operations, including different error sources. In contrast to the more common Monte Carlo based simulations used in, e.g., [9,21,24], a completely analytical method was chosen. Faulty gates are modeled as a perfect gate operation given by, e.g., one of the Pauli operators, followed by a noise channel as described in Appendix A2a. Measurements in the Z basis are also handled analytically, where the resulting density matrix is calculated for each measurement outcome and then handled separately. With this method, simulating one round of error correction is done by calculating the final states ρ_i corresponding to error syndromes S_i, e.g., (0 0 1), and then summing all of these weighted by the probability p_i of measuring syndrome S_i as

$$\rho_f = \sum_i p_i\rho_i, \tag{B1}$$

where ρ_f is the final density matrix. This method is convenient for smaller codes, as it does not involve simulating a large number of rounds to get a large enough data set. However, for larger codes with more stabilizer measurements, the problem quickly becomes prohibitively complex and expensive to calculate, even when several optimization techniques such as sparse matrix operations and caching of expensive function calls are employed. The method lends itself well to parallelization since all outcomes are calculated independently, so it is possible that considerable speedup could be achieved using a graphical processing unit.

Calculating logical error rate and code performance

For our code performance measure, we have used the logical error rate defined as the probability of an uncorrectable error being present after the error correction procedure. After an imperfect round of error correction, the density matrix will be in a mixed state. This can be taken to a pure state by performing one round of perfect, i.e., error-free, error correction. This will take correctable errors to the correct codeword, and uncorrectable errors to the opposite code word. To calculate logical error rates, we define the *reduced logical density matrix* ρ_L^{red} for pure states by

$$\rho_L^{\mathrm{red}} = \mathbf{tr}(\rho')I + \mathbf{tr}(X_L\rho')X + \mathbf{tr}(Y_L\rho')Y + \mathbf{tr}(Z_L\rho')Z, \tag{B2}$$

where X_L, Y_L, and Z_L are the logical Pauli operators defined for the given code, and X, Y, and Z are the regular Pauli matrices. By writing the error-free state $|\psi_L\rangle$ of the logical qubit as a linear combination of the logical basis vectors $|0_L\rangle$, $|1_L\rangle$ by

$$|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle, \tag{B3}$$

we can define the *reduced logical state vector* $|\psi_L^{\mathrm{red}}\rangle$ as

$$\left|\psi_L^{\mathrm{red}}\right\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{B4}$$

The logical error rate ϵ_L can then be calculated from the fidelity between ρ_L^{red} and $|\psi_L^{\mathrm{red}}\rangle$ by

$$\epsilon_L = 1 - \left\langle\psi_L^{\mathrm{red}}\middle|\rho_L^{\mathrm{red}}\middle|\psi_L^{\mathrm{red}}\right\rangle. \tag{B5}$$

APPENDIX C: TOTAL ERROR OF THE CODES

In Fig. 5 one can see the total logical and physical error as a function of the error rate p_{TQG} using the optimal n_g for each of the three QEC codes. The optimal n_g is the number of logical gates that should be applied before the QEC code to maximize the gain, and n_g was determined based on the results presented in Fig. 4. Note that when applying logical gates before the QEC protocol, the error rate threshold for beneficial error correction is higher compared to the results presented in Sec. III, which is reasonable since QEC has a substantial overhead in terms of gate operations that need to be applied as discussed in the main text.

[1] J. Wesenberg and K. Mølmer, Robust quantum gates and a bus architecture for quantum computing with rare-earth-ion-doped crystals, Phys. Rev. A **68**, 012320 (2003).

[2] A. Kinos, D. Hunger, R. Kolesov, K. Mølmer, H. de Riedmatten, P. Goldner, A. Tallaire, L. Morvan, P. Berger, S. Welinski, K. Karrai, L. Rippe, S. Kröll, and A. Walther, Roadmap for rare-earth quantum computing arXiv:2103.15743.

[3] F. Könz, Y. Sun, C. W. Thiel, R. L. Cone, R. W. Equall, R. L. Hutcheson, and R. M. Macfarlane, Temperature and concentration dependence of optical dephasing, spectral-hole lifetime, and anisotropic absorption in Eu^{3+}:Y_2SiO_5, Phys. Rev. B **68**, 085109 (2003).

[4] M. J. Zhong, M. P. Hedges, R. L. Ahlefeldt, J. G. Bartholomew, S. E. Beavan, S. M. Wittig, J. J. Longdell, and M. J. Sellars,

Optically addressable nuclear spins in a solid with a six-hour coherence time, Nature (London) **517**, 177 (2015).

[5] A. Kinos, L. Rippe, D. Serrano, A. Walther, and S. Kröll, High-connectivity quantum processor nodes using single-ion qubits in rare-earth-ion-doped crystals, Phys. Rev. A **105**, 032603 (2022).

[6] R. W. Equall, Y. Sun, R. L. Cone, and R. M. Macfarlane, Ultraslow Optical Dephasing in Eu^{3+}:Y_2SiO_5, Phys. Rev. Lett. **72**, 2179 (1994).

[7] A. Arcangeli, M. Lovrić, B. Tumino, A. Ferrier, and P. Goldner, Spectroscopy and coherence lifetime extension of hyperfine transitions in $^{151}Eu^{3+}$:Y_2SiO_5, Phys. Rev. B **89**, 184305 (2014).

[8] A. L. Alexander, J. J. Longdell, and M. J. Sellars, Measurement of the ground-state hyperfine coherence time of $^{151}Eu^{3+}$:Y_2SiO_5, J. Opt. Soc. Am. B **24**, 2479 (2007).

[9] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, and M. Müller, Assessing the Progress of Trapped-Ion Processors Towards Fault-Tolerant Quantum Computation, Phys. Rev. X **7**, 041061 (2017).

[10] D. M. Debroy, M. Li, S. Huang, and K. R. Brown, Logical performance of 9 qubit compass codes in ion traps with crosstalk errors, Quantum Sci. Technol. **5**, 034002 (2020).

[11] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S. T. Wang, L. Duan, and K. R. Brown, Simulating the performance of a distance-3 surface code in a linear ion trap, New J. Phys. **20**, 043038 (2018).

[12] T. E. O'Brien, B. Tarasinski, and L. DiCarlo, Density-matrix simulation of small surface codes under current and projected experimental noise, npj Quantum Inf. **3**, 39 (2017).

[13] J. Ghosh, A. G. Fowler, and M. R. Geller, Surface code with decoherence: An analysis of three superconducting architectures, Phys. Rev. A **86**, 062318 (2012).

[14] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, P. Neumann, and J. Wrachtrup, Quantum error correction in a solid-state hybrid spin register, Nature (London) **506**, 204 (2014).

[15] R. L. Ahlefeldt, M. J. Pearce, M. R. Hush, and M. J. Sellars, Quantum processing with ensembles of rare-earth ions in a stoichiometric crystal, Phys. Rev. A **101**, 012309 (2020).

[16] A. Steane, Space, time, parallelism and noise requirements for reliable quantum computing, Fortschr. Phys. **46**, 443 (1998).

[17] D. Gottesman, An introduction to quantum error correction and fault-tolerant quantum computation, arXiv:0904.2557.

[18] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University, New York, 2010).

[19] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[20] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, Perfect Quantum Error Correcting Code, Phys. Rev. Lett. **77**, 198 (1996).

[21] R. Chao and B. W. Reichardt, Quantum Error Correction with Only Two Extra Qubits, Phys. Rev. Lett. **121**, 050502 (2018).

[22] A. M. Steane, Error Correcting Codes in Quantum Theory, Phys. Rev. Lett. **77**, 793 (1996).

[23] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).

[24] Y. Tomita and K. M. Svore, Low-distance surface codes under realistic quantum noise, Phys. Rev. A **90**, 062320 (2014).

[25] P. W. Shor, Fault-tolerant quantum computation, arXiv:quant-ph/9605011.

[26] A. M. Steane, Active Stabilization, Quantum Computation, and Quantum State Synthesis, Phys. Rev. Lett. **78**, 2252 (1997).

[27] A. M. Steane, Fast fault-tolerant filtering of quantum codewords, arXiv:quant-ph/0202036.

[28] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Towards Practical Classical Processing for the Surface Code, Phys. Rev. Lett. **108**, 180501 (2012).

[29] K. M. Svore, A. W. Cross, I. L. Chuang, and A. V. Aho, A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing, Quantum Inf. Comput. **6**, 193 (2006).

[30] A. Kinos, L. Rippe, S. Kröll, and A. Walther, Designing gate operations for single-ion quantum computing in rare-earth-ion-doped crystals, Phys. Rev. A **104**, 052624 (2021).

[31] A. Walther, L. Rippe, Y. Yan, J. Karlsson, D. Serrano, A. N. Nilsson, S. Bengtsson, and S. Kröll, High-fidelity readout scheme for rare-earth solid-state quantum computing, Phys. Rev. A **92**, 022319 (2015).

[32] B. Lauritzen, S. R. Hastings-Simon, H. De Riedmatten, M. Afzelius, and N. Gisin, State preparation by optical pumping in erbium-doped solids using stimulated emission and spin mixing, Phys. Rev. A **78**, 043402 (2008).

[33] A. M. Souza, G. A. Álvarez, and D. Suter, Robust Dynamical Decoupling for Quantum Computing and Quantum Memory, Phys. Rev. Lett. **106**, 240501 (2011).

[34] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

[35] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, Proc. Natl. Acad. Sci. USA **114**, 7555 (2017).

[36] S. J. Devitt, W. J. Munro, and K. Nemoto, Quantum error correction for beginners, Rep. Prog. Phys. **76**, 076001 (2013).

[37] E. Knill, Quantum computing with realistically noisy devices, Nature (London) **434**, 39 (2005).