Subspace Randomized Benchmarking Prediction Protocol for the Average Gate Fidelity of Multi-Qubit Devices

Daniël Veldhuizen - 4732359

Bachelor's Thesis
BSc Applied Physics and Applied Mathematics
Faculties of Applied Sciences and EEMCS
Delft University of Technology

Delft, June 24, 2021
Supervisors: prof. dr. L.M.K. Vandersypen,
dr. M.P.T. Caspers
Daily supervisor: dr. M.F. Russ

Contents

Abstract

Quantum computers promise an exponential speed-up over their classical counterparts for certain tasks relevant to various fields including science, technology, and finance. To unlock this potential, the technology must be scaled up and the errors at play must be reduced. As developments in scalable quantum computation devices advance, the demand for scalable benchmarking techniques that are able to reliably assess the fidelity – the complement of the error rate – of a device has increased significantly. Randomized benchmarking offers a single, concise number that reflects the average fidelity of multi-qubit operations performed on a quantum device. While this method is robust against state preparation and measurement errors, it still suffers from scalability issues. In this thesis, we present a protocol that efficiently predicts the multi-qubit fidelity obtained from randomized benchmarking by only benchmarking single- and two-qubit subspaces, greatly increasing the scalability. The protocol uses simultaneous randomized benchmarking with the aim of catching cross-talk effects while at the same time reducing the number of required benchmarking sequences. We have run numerical simulations of the protocol under two noise models, one depolarizing and one dephasing, to verify its performance. The results of these noisy simulations are promising and suggest that our protocol could offer a valuable tool on the road to developing large-scale quantum computers.

1 | Introduction

The silicon transistor gave birth to a technological revolution that has changed the way we interact with the world and with each other. While classical computing power has increased exponentially over the years and has become widely available, certain classes of problems are still difficult or infeasible to solve. The emergence of quantum computing technology promises the ability to solve many of these problems efficiently [1]. A few potential applications are simulating complex (quantum) systems such as chemical reactions, factoring large integers, and highly efficient search algorithms [2]. The development of a commercially viable quantum computer may lead to advances in the fields of materials science, chemistry, pharmacy, cryptography, and many others. The quantum computer aims to surpass classic computers in certain tasks by exploiting the principles of quantum mechanics [2]. Instead of storing information in bits – either 0 or 1 – the information is stored in qubits. A qubit is any 2-level system with available states $|0\rangle$ and $|1\rangle$. What distinguishes a quantum computer is the superposition principle: any normalized sum of the two basis states is again a valid state. The other quantum mechanical phenomenon crucial to quantum computation is entanglement: two states may be coupled in such a way that correlations between their measurements are stronger than is classically possible. There are two major obstacles between the current state-of-the-art devices and the quantum computer of the future. The first is scaling up the number of qubits. To tackle any problem of practical interest the number of perfectly controlled noiseless qubits needs to be increased. The words perfectly controlled and noiseless are important and bring us to the second obstacle: error mitigation. Any system of qubits experiences undesired coupling among qubits and with the environment. Additionally, it is practically infeasible to reach perfect control, which means almost all operations are imperfectly performed. With noisy qubits the estimated required number of qubits for useful applications increases even more because extra qubits are needed to implement error correction schemes [3]. Instead of using the error rate, the term fidelity is used throughout the literature. The fidelity is simply the complement of the error probability, i.e. the success rate.

Although distinct, it is clear that both of these problems must be tackled in tandem. Many candidates for a scalable quantum computer are currently being researched. Some noteworthy examples include superconducting qubits [4], trapped ions [5], nitrogen-vacancy centers [6] and silicon-based quantum dots [7]. This report focuses on a Si/SiGe-based 6-qubit spin qubit setup developed in the VandersypenLab at QuTech. This device performs a calculation by preparing a certain state, performing operations on this state – thereby changing it – and measuring the resultant state. Devices that operate in this fashion are called gate-based quantum computers, where the term ’gate’ is used to refer to a performed operation.

The need to increase fidelity called for the development of so-called benchmarking or tomography protocols. These protocols can be performed on a device in order to quantify the fidelity of a specific gate, set of gates, or average gate. Quantum process tomography [8], for example, enables complete mathematical characterization of a quantum process. This type of protocol has several important drawbacks. The resources required to perform the protocol scale exponentially with the number of qubits. Additionally, quantum process tomography is unable to distinguish errors caused by the application of a gate from the errors inherent to state preparation and measurement (SPAM). Randomized benchmarking (RB) offers a convenient alternative approach by providing a single, concise benchmarking metric [9]. Advantages of randomized benchmarking include its robustness against SPAM errors, relatively low required resources, and the effectivity in case of low error rates. Despite these advantages, it is still challenging to scale up randomized benchmarking to higher numbers of qubits. In this report we present a protocol that can be used to predict the multi-qubit fidelity obtained from multi-qubit randomized benchmarking using the results of single- and two-qubit randomized benchmarking. Although there a several more advanced variants of the randomized benchmarking protocol which aim to produce more descriptive metrics such as interleaved randomized benchmarking [10] and character randomized benchmarking [11], we shall not pursue these here.

In chapter 2 of this report, a brief introduction to quantum information theory is provided. Additionally, the framework behind randomized benchmarking is laid out. The main body of the work is contained in chapter 3, which is followed by a discussion in chapter 4 that also provides additional ideas to advance further. The thesis concludes in chapter 5 with a summary.

This research is done in the context of the Bachelor Applied Physics and Applied Mathematics at the Delft University of Technology to obtain the degree of Bachelor of Science.

2 | Theory

2.1. Introduction to Quantum Information Theory

2.1.1. Density operator formalism

We want to describe the state of a system that is a statistical ensemble of quantum states. To make it clear what we mean, suppose we have a state $|0\rangle$ and a state $|1\rangle$, with probabilities of p_0 and p_1 respectively that the system is one of the aforementioned states. Then the system is in a statistically mixed state consisting of the pure states $|0\rangle$ and $|1\rangle$. The term pure is used to clarify the distinction between pure and mixed states. In general we may have a set of states $\{|i\rangle\}_{i\geq 0}$ with corresponding probabilities $\{p_i\}_{i\geq 0}$, together composing an ensemble of states. Such a system may be described by a density operator ρ which we define as

$$\rho \equiv \sum_i p_i |i\rangle \langle i| . \tag{2.1}$$

The density matrix satisfies the following properties [2]

$$\rho^\dagger = \rho \tag{2.2}$$

$$\text{Tr}(\rho) = \sum_i \rho_{ii} = 1. \tag{2.3}$$

Additionally, any density matrix describing a pure state has the property that $\text{Tr}(\rho^2) = 1$ while any mixed state satisfies $\text{Tr}(\rho^2) < 1$.

Bloch sphere

A qubit is a two-level system described by a pair of vectors $\{|0\rangle, |1\rangle\}$ spanning a two-dimensional Hilbert space. One of the principles of quantum mechanics is that any state can be written as a superposition of the basis states. Additionally any state should be normalized such that $\langle i|i\rangle = 1$ for any i. Using the defined basis and the normalization requirement we find that any state may be written as

$$|i\rangle = \cos(\theta/2) |0\rangle + \sin(\theta/2)e^{i\phi} |1\rangle , \tag{2.4}$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$. It is easily checked that this state stratifies the normalization condition, thus describing a physical state. The idea is now to interpret θ and ϕ as angles describing the surface of a sphere. This idea allows us to visualize all possible states of a two-level system as unique points on the surface of a unit sphere: the Bloch sphere, as shown in Figure 2.1. Equivalently, each physical state has an associated vector that points from the center of the Bloch sphere to a point on its surface. Such a vector is called a Bloch vector

$$\vec{a} = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta)). \tag{2.5}$$

Moreover, the interior of the Bloch sphere is of interest too, describing all possible mixed states. It turns that any density matrix ρ describing a 2-level system may be expanded in terms of these matrices as

$$\rho = \frac{1}{2}\big(\mathcal{I} + \vec{a}\cdot\vec{\sigma}\big) = \frac{1}{2}\big(\mathcal{I} + a_x X + a_y Y + a_z Z\big) = \frac{1}{2}\begin{pmatrix} 1 + a_z & a_x - ia_y \\ a_x + ia_y & 1 - a_z \end{pmatrix}, \tag{2.6}$$

where the Pauli matrices are given by

$$P_0 \equiv \mathcal{I}, \quad X = P_1 \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = P_2 \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = P_3 \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.7}$$

The eigenvalues of the single-qubit density matrix in equation 2.6 are $\frac{1}{2}(1 \pm |\vec{a}|)$. Due to physicality constraints we must have that any density operator is positive semi-definite (i.e. all its eigenvalues λ satisfy $\lambda \geq 0$ we find immediately that $|\vec{a}| \leq 1$. We have found that all pure states lie on the surface of the Bloch sphere while all mixed states lie in its interior, with the length of the Bloch vector revealing the degree of mixture.

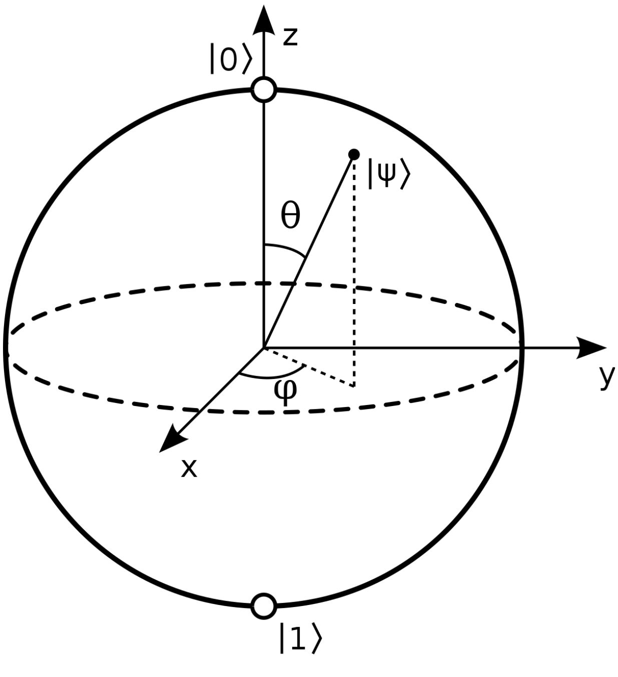

Figure 2.1: Illustration of an arbitrary two-qubit state of the Bloch sphere. Each single-qubit pure state is represented by a point on the surface of the sphere according to equation 2.5. Mixed states are then in the interior of the sphere. This visualization is helpful in understanding single-qubit states and quantum operations, which correspond to rotations. The Bloch sphere visualization does not hold for multi-qubit states. These require a higher-dimensional sphere.

POVM

Here we present a brief overview of the Positive-Operator Valued Measure (POVM) in the context of quantum information theory. The POVM plays the role of an observable, i.e. a physical and measurable quantity. We will consider the case in which the Hilbert space, as well as the number of elements of POVM, are finite. A POVM is a set of positive semi-definite matrices $\{F_j\}$ on the relevant Hilbert space that sum to the identity

$$\sum_j^n F_j = \mathcal{I}. \tag{2.8}$$

In quantum mechanics each POVM element F_j is linked to the outcome of measurement j and the probability to find outcome j is given by

$$p(j) = Tr(F_j \rho). \tag{2.9}$$

We note that in the special case that ρ is a pure state, i.e. $\rho = |\psi\rangle \langle\psi|$ we find

$$p(j) = Tr(F_j |\psi\rangle \langle\psi|) = \langle\psi|F_j|\psi\rangle\,, \tag{2.10}$$

due to the construction of the density matrix. The POVM formalism provides a mathematical description of a quantum mechanical measurement giving rise to collapse of the wavefunction.

2.1.2. Quantum operations

A unitary operator action on a state transforms a pure state into a new pure state. For any unitary matrix U the mapping is given by $|i\rangle \xrightarrow{U} U|i\rangle$. In the single qubit case, a unitary operation rotates the Bloch vector around some axis. The information in our system is represented by the density operator ρ. Changing the state of our system then requires a transformation of the density operator. Similar to the kets, the density operators are elements of some Hilbert space $\mathcal{H}$. For some $\rho, \rho' \in \mathcal{H}$ a linear map $\Lambda : \rho \rightarrow \rho'$ that acts on the space of density operators is called a superoperator. For the case of a unitary operation, we obtain

$$\rho = \sum_i p_i |i\rangle \langle i| \xrightarrow{\Lambda} \sum_i p_i U |i\rangle \langle i| U^\dagger = U\rho U^\dagger = \rho', \tag{2.11}$$

where † denotes Hermitian transposition. In quantum information theory, a superoperator acting on a density matrix is usually called a quantum channel or quantum map. In principle the presented formalism works for arbitrary d-level systems (also called qudits), however, we restrict ourselves only to systems of qubits. This means $\mathcal{H} = \mathcal{H}_n$ where $\mathcal{H}_n$ is a 2^n-dimensional Hilbert space whose basis states describe a system of n qubits.

Since the goal is to describe a physical process we must impose some additional constraints on these operations. Fortunately, there are only two constraints that need to be satisfied. The first is that after the application of the map, the resultant density operator needs to have non-negative probabilities for measuring the eigenstate of an arbitrary observable. This constraint means that that the map is required to be completely positive (CP). Secondly, we demand that probabilities must be conserved, i.e. the operator must be trace-preserving (TP). In general, this second condition may be loosened to take into account leakage into the environment. However, we will not take leakage errors into consideration in this report and thus only consider CPTP maps.

Kraus Decomposition

Any CP map can be written in the as [2]

$$\Lambda[\rho] = \sum_{i=1}^{N} K_i \rho K_i^{\dagger}, \tag{2.12}$$

for a certain $N \leq d^2$, $d = 2^n$ with 2^n the dimension of the Hilbert space in question. This is called the Kraus representation of the map. The K_i are called Kraus operators. In order to include the TP constraint as well we must have that

$$\sum_{i=1}^{N} K_i K_i^{\dagger} = \mathcal{I}, \tag{2.13}$$

which is the completeness condition. The decomposition into Kraus operators is in non-unique in general. Additionally the Kraus operators need not be invertible, unitary or Hermitian.

Pauli completeness

In equation 2.7 we have introduced the Pauli matrices on a single qubit and we have shown that every single qubit (two-level system) density matrix can be decomposed into these matrices. In fact, the same holds for any n-qubit density matrix. Therefore we expand the Pauli matrices to n-qubit spaces by considering tensor products of single-qubit Pauli matrices. For $n \geq 1$ there are 4^n generalized Pauli matrices given by

$$\mathbf{P}_n = \{\mathcal{I}, X, Y, Z\}^{\otimes n}. \tag{2.14}$$

The notation $P_i \otimes P_j$ is often abbreviated as $P_i P_j$, where P_i and P_j are arbitrary single qubits Pauli matrices. We adopt this convention in the remainder of this thesis.

Quantum Errors

Implementing quantum operations on an engineered device is challenging and prone to errors. There a various mechanisms within real live systems which may cause undesired behavior. This noise may in general be quite complex: it may both operator- and time-dependent. Furthermore, in a real device, these errors are often correlated, i.e. each error can depend on all of the previous errors. In this report, we make the explicit assumption that there is no time dependence and that the errors are uncorrelated. We additionally do not make an effort to describe gate-dependent noise. However, we will later comment on why this is not required to obtain a valid benchmark on systems with said gate-dependent noise. When considering quantum noise we distinguish between two main types of errors.

Stochastic noise

Stochastic noise is – as implied – of a stochastic nature. This type of noise reduces the amount of available information. We may describe this type of error, without loss of generality, as a probabilistic projection onto an n-qubit Pauli basis. It may be seen as a list of tuples (p_i, P_i), where p_i is the probability that the state being acted on is projected onto P_i and $\sum_i p_i = 1$. We consider two types of noise channels that are of this type.

Depolarising channel

This first stochastic noise channel we consider is the depolarising channel

$$\Lambda_{\text{dep}}[\rho] = \alpha\rho + \frac{1-\alpha}{d}\mathcal{I}, \tag{2.15}$$

where α is a scalar and $\mathcal{I}$ is the identity matrix of the same dimension as the density matrix. We see that the superoperator acts on a density matrix and returns another density matrix. Intuitively this superoperator may be interpreted to do the following: do nothing with probability α, i.e. leave the state ρ intact, and with probability $1-\alpha$, transform the state ρ into the completely mixed state $\mathcal{I}$. In reality, the state is brought into a statistical mixture of these two options and a measurement will yield either result with the respective probability. In essence, the depolarizing channel destroys all information in the system with probability $1-\alpha$. A depolarising channel may be visualized as a uniform shrinking of the Bloch sphere, or more precisely as shrinking of the length of the Bloch vector.

Dephasing channel

A stochastic noise channel that is less extreme than complete depolarization is the dephasing channel. This channel represents the loss of all phase information only with probability $1-\alpha$. One possible Kraus decomposition is given by

$$K_0 = \sqrt{1 - \frac{1-\alpha}{2}}\mathcal{I}, \qquad K_1 = \sqrt{\frac{1-\alpha}{2}}Z. \tag{2.16}$$

Visually, this operation projects the Bloch vector onto the Z-axis of the Bloch sphere, thus deforming the Bloch sphere into a Bloch ellipsoid.

Unitary noise

In contrast to stochastic errors, unitary errors do not reduce the amount of information in the system. On the Bloch sphere this means that the operation does not shrink the Bloch sphere. Instead, the vector is rotated over some axis. These types of errors may therefore be encountered when gate implementation is imperfect. A very common example is an overrotation when implementing a Pauli X gate. We model this overrotation due to imperfect control using a unitary noise channel.

2.1.3. Superoperator formalism

One major drawback of working with superoperators of this form is that the composition of arbitrary maps is tedious and not necessarily insightful. An alternative way of working with quantum maps is to use a different representation. One of the most convenient representations is the Pauli Transfer Matrix (PTM). The PTM representation R_Λ of a quantum channel Λ is defined as

$$(R_\Lambda)_{ij} = \frac{1}{d}\text{Tr}(P_i\Lambda[P_j]), \tag{2.17}$$

where P_k are the Pauli matrices (including the identity) on n qubits. PTMs have several useful properties, the most important of which is that their composition is equal to their matrix multiplication. The PTM can be understood as acting on a density operator ρ represented as a vector in a 4^n-dimensional space[12]. Essentially the PTM representation is a mapping of Pauli operators onto Pauli operators.

2.2. Randomized Benchmarking

Randomized benchmarking is a protocol to benchmark the quality of (average) gate operations. It provides a single number that can be related to the average fidelity of the average applied operation. Here we introduce a basic version of the randomized benchmarking protocol [13]. We start by introducing the Clifford group $\mathcal{C}_n$ on n qubits. The Clifford group on n qubits is defined as the normalizer in the unitaries of the Pauli group on n qubits $\mathcal{P}_n$. In short, it maps Paulis to Paulis and is defined as

$$\mathcal{C}_n = N_{\mathcal{U}_{2^n}}(\mathcal{P}_n) = \{U \in \mathcal{U}_{2^n} \,|\, U\mathcal{P}_nU^\dagger = \mathcal{P}_n\}. \tag{2.18}$$

Here $\mathcal{U}_{2^n}$ is the unitary group acting on a space of dimension 2^n. The Pauli group on n qubits is formed by enlarging the set of n-qubit Paulis to the same set including scalar multiples $\{1, -1, i, -i\}$ of all elements. This ensures that the new set becomes a group which we denote as $\mathcal{P}_n$

$$\mathcal{P}_n = \{\mathcal{I}, X, Y, Z\}^{\otimes n} \otimes \{\pm 1, \pm i\}. \tag{2.19}$$

One reason for using the normalizer of the Pauli group is to make calculations simpler. The approach to many problems in quantum information theory is to decompose the operations into Pauli operators. Working with the normalizer of this group is therefore natural.

We will now present the basic protocol in detail. First, we initialize an array M of sequence lengths m. How this choice is made will be explained in more detail in Chapter 3. Now, for each fixed $m \in M$, we continue by constructing K_m sequences of m uniformly random Clifford elements. Throughout this report we choose $K_m = K \; \forall m$ where we call K the number of seeds. The resulting sequence is then denoted as $\{C_{i_j}\}_{0 \leq j \leq m}$. Here the index i runs from 1 to K. We may efficiently calculate the inverse gate $C_{i_{m+1}}$ of the entire sequence according to the Gottesmann-Knill theorem [14]. The theorem states that any circuit composed of only gates from the Clifford group can be efficiently simulated (i.e. in polynomial time) on a classical computer. By appending the calculated inverse gate to the sequence we obtain the identity operation $\mathcal{I}$ as the composition of the entire sequence. However, we must realize that in reality, each Clifford operation has some error Λ_{i_j} associated with it. Note that the error channel depends in general both on the sequence it occurs in as well as the position it appears in. The resulting operation of applying the entire sequence (including the inverse gate), denoted $S_{\boldsymbol{i_m}}$, is given by

$$S_{\boldsymbol{i_m}} = \bigcirc_{j=1}^{m+1} \big(\Lambda_{i_j} \circ C_{i_j}\big), \tag{2.20}$$

where $\boldsymbol{i_m} = (i_1, ..., i_m)$, i_{m+1} is uniquely determined by $\boldsymbol{i_m}$ and $\circ$ denotes composition. For fixed i, $\boldsymbol{i_m}$ is an m-tuple of indexes that associates a certain gate with each position in the gate sequence of seed number i. This assumes a bijection between the index set and the Clifford group. The final Clifford is determined by all of the preceding Clifford, hence i_{m+1} is determined by $\boldsymbol{i_m}$. The errors have the effect that the composition of the total sequence is not equal to the identity operation anymore. For each of the K sequences the survival probability of the initial state is given by

$$(P_s)_m = \text{Tr}(E_\psi S_{\boldsymbol{i_m}}[\rho_\psi]). \tag{2.21}$$

Here E_ψ is the POVM element that accounts for measurement errors and ρ_ψ is the initial state. For an ideal measurement we would simply have $E_\psi = |\psi\rangle\langle\psi|$; real quantum computers suffer from SPAM errors. We may now calculate the average sequence fidelity by averaging over all sequences of a fixed length yielding

$$F_{\text{seq}}(m, |\psi\rangle) = \text{Tr}(E_\psi S_K[\rho_\psi]), \tag{2.22}$$

where we have

$$S_{m,K} = \frac{1}{K}\sum_{\boldsymbol{i_m}} S_{\boldsymbol{i_m}}. \tag{2.23}$$

Here the sum is over all K sequences of length m. This last superoperator can be seen as the the average sequence operation. The powerful result used in randomized benchmarking is that we may fit the our measured average sequence fidelity to the model

$$F_{\text{seq}}(m, |\psi\rangle) = A\alpha^m + B. \tag{2.24}$$

Conveniently, the parameters A and B absorb all SPAM errors. The decay parameter α now contains information about the average error occurring in the process of applying the gates. We may restate the depolarization parameter in terms of the average error rate (or infidelity) r using the following relation [15]

$$r = 1 - \alpha - \frac{1-\alpha}{2^n} = \frac{2^n - 1}{2^n}(1-\alpha), \tag{2.25}$$

where n is the number of qubits.

2.2.1. Intuitive understanding

We will now attempt to justify some of the steps taken to derive the randomized benchmarking protocol. Earlier we have introduced the depolarization superoperator

$$\Lambda[\rho] = \alpha\rho + \frac{1-\alpha}{2^n}\mathcal{I}. \tag{2.26}$$

Let us now suppose that we have a sequence of m arbitrary gates where the error channel of each gate is a depolarizing channel with parameter α. Note that we are considering arbitrary gates, not necessarily Clifford gates. Intuitively it makes sense that a depolarizing channel as described above should commute with any other gate. Additionally, it seems like the composition of two α-depolarising channels should results again in a depolarizing channel with depolarizing parameter $\alpha' = \alpha^2$. In fact, it is trivial to show this last claim is true

$$\Lambda[\Lambda[\rho]] = \alpha(\alpha\rho + \frac{1-\alpha}{2^n}\mathcal{I}) + \frac{1-\alpha}{2^n}\mathcal{I} = \alpha^2\rho + \frac{1-\alpha^2}{2^n}\mathcal{I}. \tag{2.27}$$

Generalizing to our sequence of m gates we find that the combined error channel of the sequence is an α^m-depolarizing channel. Assuming that we start in the ground state and that the applied sequence (without error channels) amounts to the identity, as is the case in our protocol, we can then measure the state after the sequence using the standard basis. It is straightforward to show that this directly yields the RB formula introduced in equation 2.24

$$Tr(\rho\Lambda^m[\rho]) = \alpha^m + \frac{1-\alpha}{2^n} = \frac{2^n-1}{2^n}\alpha^m + \frac{1}{2^n} = A\alpha^m + B, \tag{2.28}$$

since we simply take the trace over a sum of scalar multiples of trace one matrices. What we find is that the probability of success (fidelity), i.e. finding the ground state after applying the sequence, decays exponentially with the sequence length m and decay parameter α.

This derivation is of course not valid for arbitrary error channels other than the depolarizing channel. Fortunately, it turns out that there is a mathematical result that saves our approach. When the gates composing the sequence are uniformly random samples from the Clifford group $\mathcal{C}_n$ the error on each gate behaves on average as a depolarizing channel with a corresponding depolarization parameter. This particularly useful result allows us to still use a measured decay parameter to benchmark the average fidelity.

2.2.2. Formal explanation

In rigorous mathematical language, we are taking the average over a finite group (here the Clifford group) of a certain quantum channel Λ. In the case of randomized benchmarking the channel that is being averaged over is actually the average of error channels Λ_{i_j}, denoted by $\bar{\Lambda}$. The assumption here is that $\Lambda_{i_j} - \bar{\Lambda}$ is small [15][13]. This averaging operation is called a twirl in the literature (denoted $\mathcal{W}$) and is defined by the following operation

$$\mathcal{W}_G(\bar{\Lambda}) = \frac{1}{|G|}\sum_{\mathcal{U}\in G}\mathcal{U}^\dagger \circ \bar{\Lambda} \circ \mathcal{U}. \tag{2.29}$$

Here G denotes a general finite group with group order (size) $|G|$. We sum over all group elements $\mathcal{U}$, where $\mathcal{U}[\rho] = U\rho U^\dagger$ is its adjoint representation. In the PTM representation, the twirl operation is given by

$$\mathcal{W}_G(R_{\bar{\Lambda}}) = \frac{1}{|G|}\sum_{\mathcal{U}\in G} R_{\mathcal{U}}^\dagger R_{\bar{\Lambda}} R_{\mathcal{U}}. \tag{2.30}$$

In the case of the Clifford group the resulting channel is a depolarizing channel [16]. Furthermore, it can be shown that twirling over the entire unitary group yields the same result as twirling over the Clifford group. Note that twirling over the entire unitary group amounts to taking a weighted integral over the entire infinite unitary group using the appropriate uniform (Haar) measure. What may be concluded is that randomized benchmarking results in a doubly averaged fidelity, once over the applied errors and once over all unitary operations.

2.2.3. Confidence bounds

One of the great advantages of randomized benchmarking is the ease with which the confidence bound on the result can be extracted. Similarly to the fidelity, the confidence bound may be found directly from the fit [15]. This means that given enough seeds and a sufficient number of sequence lengths the obtained confidence bound may be relatively tight. This is all under the assumption that the observed decay after performing the twirl operation does not deviate (significantly) from exponential decay. When such deviations are visible they must be treated accordingly, e.g. as in done in [16]. For the randomized benchmarking approach taken here – considering that it is mostly based on standard RB – there are no such deviations expected. We will therefore leave a further discussion with regard to confidence bounds to the Discussions section of this report.

2.3. Predicting multi qubit fidelity

For the case of the depolarizing channel introduced in equation 2.15 a possible Kraus decomposition is

$$K_0 = \sqrt{\alpha + \frac{1-\alpha}{d^2}}\mathcal{I}, \tag{2.31}$$

$$K_i = \sqrt{\frac{1-\alpha}{d^2}}P_i, \tag{2.32}$$

where $\mathcal{I}$ is the $d \times d$ identity matrix and P_i are the Pauli matrices matrices on n qubits. Note that in this case the Pauli matrices are regarded as the full set bar the identity element, making it a set of $4^n - 1$ elements.

In the randomized benchmarking protocol we apply sequences of Clifford gates. In practice, these Clifford gates are decomposed into a certain set of basis operations native to the setup, which are then applied. In our analysis we assume that each of these underlying basis operations is composed with a depolarizing channel. What is important to note here is that these basis operations do not necessarily act on the full space but may act only on a certain subspace. We must therefore know how a depolarizing channel acts in a subspace. We may construct such a channel by only considering Paulis in equation 2.32 that are non-identity in the relevant subspace [17]

$$\Lambda[\rho] = \alpha\rho + \frac{1-\alpha}{d_s^2}\left(\rho + \sum_{P \in S}(P\rho P)\right), \tag{2.33}$$

with d_S the dimension of the subspace. For a depolarizing channel in a two-qubit space acting solely on qubit 0 we thus consider all Paulis except $\{II, IX, IY, IZ\}$. Since these depolarizing channels commute with the basis gates we may construct the error channel associated with the average Clifford gate – consisting of n basis operations – as

$$\bar{\Lambda}_C[\rho] = \Lambda_{\text{dep},1}[\rho] \circ ... \circ \Lambda_{\text{dep},n}[\rho], \tag{2.34}$$

in which $\Lambda_{\text{dep},i}$ denotes the depolarizing channel associated with the i^{th} composing basis operation. We now conveniently switch to the PTM representation. For a depolarising channel in a subspace the PTM is diagonal with the diagonal elements determined as

$$(R_\Lambda)_{ii} = \begin{cases} \alpha & \text{if } \exists P \in S \text{ s.t. } [P_i, P] \neq 0 \\ 1 & else \end{cases}. \tag{2.35}$$

To find the composition of the maps $\Lambda_1 ... \Lambda_n$ we multiply the corresponding PTMs

$$R_{\Lambda_C} = R_{\Lambda_N} ... R_{\Lambda_1}. \tag{2.36}$$

Here we find, unsurprisingly, that any finite composition of depolarising maps is again a depolarising map. In order to find the associated α parameter we compute [16]

$$\alpha_C = \frac{Tr(R_{\Lambda_C}) - 1}{d^2 - 1}. \tag{2.37}$$

3 | Protocol and Results

3.1. Device

Quantum computers based on semiconductor spin qubits are promising due to their small footprint, the prospects for scaling, and their compatibility with semiconductor fabrication techniques [18]. Here we present a brief overview of the physics behind the 6-spin-qubit device. Using state-of-the-art semiconductor growth techniques it is possible to create Si/SiGe heterostructures [19]. These heterostructures are similar to the ones used for a conventional transistor. Such a heterostructure allows the formation of a two-dimensional electron gas close to the interface, confining an electron in two dimensions. Additionally, an electrostatic potential can be tuned in such a way that an electron is trapped in quantum dot [20]. Similar to an atom, the trapped particle resides in a quantized state which can be controlled via external signals [21]. For spin qubits, the information is encoded in the spin degree of freedom of the trapped particle. However, in order to have non-degenerate spin states for all qubits an external magnetic field is applied and together with a micromagnet induces a Zeeman splitting that ideally lifts all degeneracies. The result is a linear array of spin qubits which are coupled as displayed in Figure 3.1. One of the more common types of information loss in the device is spin decoherence, which can be approximately modeled as a dephasing channel. The most notable causes of decoherence errors are the hyperfine interaction and charge noise via spin-orbit coupling. A scanning electron microscope (SEM) image of such a heterostructure-based spin qubit device is shown in Figure 3.2.

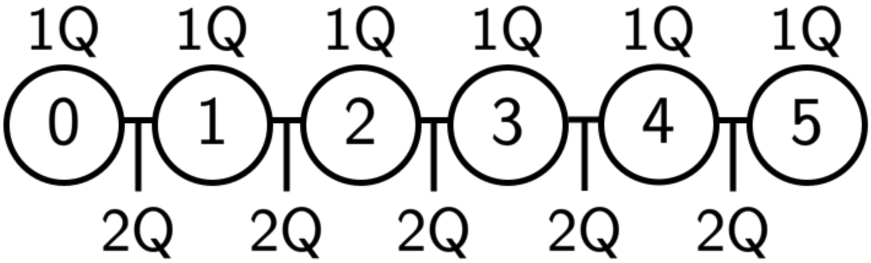

Figure 3.1: Connectivity map of the linear 6 spin-qubit setup. Each qubit can be coupled to its nearest neighbour which enables performing two-qubit gates on such pairs. Single-qubit operations can additonally performed on each individual qubit.

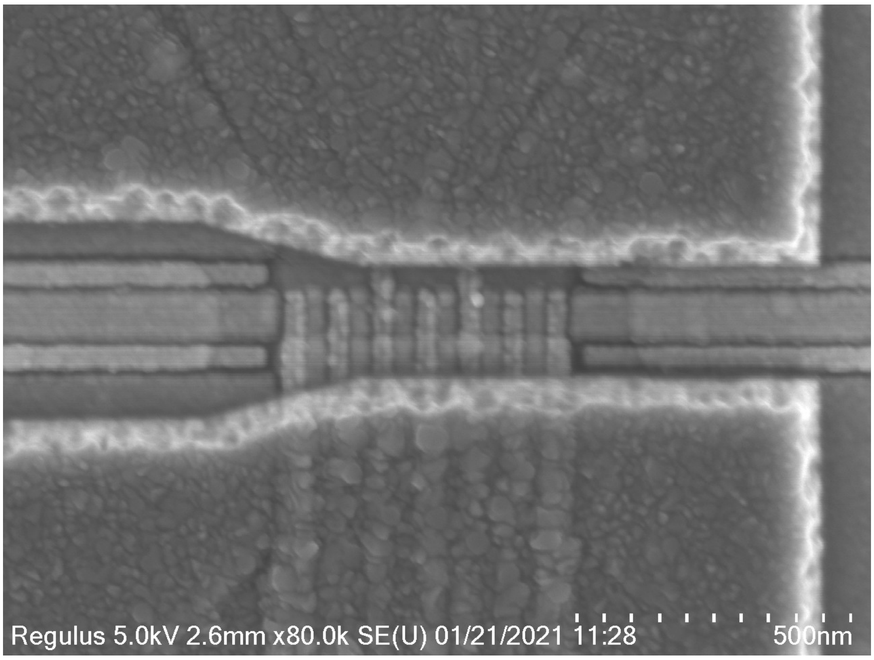

Figure 3.2: SEM image of a 6 dot device from the Vandersypen lab.

When engineering a gate-based quantum computing device one of the main priorities is realizing high single- and two-qubit gate fidelities. Extensive work has been done to characterize these fidelities for this device in detail [10]. In this report we instead aim to benchmark the multi-qubit average gate fidelity of the device. This is relevant since the ultimate goal of increasing the number of qubits is the ability to perform more complex and with that hopefully more powerful operations. Certainly, the average fidelity of these operations must be high enough for error correction to be effective. To be able to benchmark this fidelity we must first establish the possible gates we can perform on the device. The protocol is implemented and numerically verified using IBM's Qiskit Python package [22]. Due to implementation limitations in Qiskit, we choose not to execute the protocol in terms of gates native to the device. Instead, we use the default set of single-qubit gates for IBM quantum devices:

$$U_1(\lambda) = \begin{pmatrix} 1 & 0 \\ e0 & e^{i\lambda} \end{pmatrix}, \tag{3.1}$$

$$U_2(\gamma, \lambda) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\lambda} & e^{i(\gamma+\lambda)} \end{pmatrix}, \tag{3.2}$$

$$U_3(\beta, \gamma, \lambda) = \frac{1}{\sqrt{2}} \begin{pmatrix} \cos(\beta/2) & -e^{i\lambda}\sin(\beta/2) \\ e^{i\lambda}\sin(\beta/2) & e^{i(\gamma+\lambda)}\cos(\beta/2) \end{pmatrix}, \tag{3.3}$$

where $U_2(\gamma, \lambda)$ is able to perform arbitrary rotations about the $X + Y$ axis of the Bloch sphere and $U_3(\beta, \gamma, \lambda)$ can be used to perform rotations about an any axis. The $U_1(\lambda)$ gate acts a single-qubit revolution around the Z axis and is usually implemented virtually using software. Evidently, if the protocol is to be performed on the device an additional transpilation layer is needed which translates U_1, U_2 and U_3 gates to the native gates of the device. Implementing such a transpilation layer is beyond the scope of this presentation. As the basic two-qubit gate we use a controlled phase gate, or CZ gate, defined as

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{3.4}$$

The gate acts on two qubits simultaneously, one is the control qubit and the other is the target qubit. Such a gate describes a conditional process where the phase is flipped if the control qubit is in the $|1\rangle$ state.

3.2. Protocol

Qiskit is used to simulate randomized benchmarking experiments on the qubit setup. The aim of the protocol is to use the mathematical approach outlined in section 2.3 to estimate the average gate fidelity of the full 6-qubit array from the single-qubit and two-qubit fidelities. There are few constraints the protocol must adhere to. The primary constraint is that the 6-qubit spin qubit setup allows coupling neighboring qubits only **(1)**. To perform operations on non-neighboring qubits the protocol must perform SWAP operations. As the name suggests a SWAP operation swaps the state of two (in this case neighboring) qubits. From the mathematical approach is it apparent that we must benchmark every single qubit and each pair of qubits at least once **(2)**. To increase the robustness of the method we would like to use simultaneous randomized benchmarking as much as possible **(3)**. With this approach we expect to catch crosstalk effects. Another advantage of using simultaneous randomized benchmarking is to reduce the number of experiments required to predict the multi-qubit average gate fidelity. A link to the code can be found in appendix B. We first present an overview of the entire benchmarking protocol:

Protocol overview

- Use standard randomized benchmarking to find the average fidelity of every single qubit and pair of qubits under the constraints and objectives **(1)**, **(2)** and **(3)**.
- From these fidelities, calculate the average fidelity of the underlying basis gates (U_1, U_2, U_3 and CZ) according to the theory outlined in section 2.3.
- Using the same theory and the newly obtained average basis gate fidelities predict the multi-qubit average gate fidelity.

An in-depth explanation of the protocol along with its implementation is presented in the following sections.

3.2.1. Randomized Benchmarking in Qiskit

This section aims to clarify how randomized benchmarking can be numerically simulated in Qiskit. Many useful implementations (classes/functions) are available natively from within Qiskit's open-source library. Since we want to perform multiple randomized benchmarking experiments we first construct an Experiment class.

Experiment Class

The Experiment class consists of all variables and methods relating to a single randomized benchmarking experiment. In order to initialize an experiment we need the following parameters

- The number of qubits `nQ` on which the experiment is performed.
- A dictionary `rb_opts` containing details about the experiment:
  - An array `length_vector` containing the lengths of the randomized benchmarking sequences to be performed. This array is called M in the Theory.
  - The number of seeds `nseeds` referred to as K in the Theory.
  - A nested list `rb_pattern` specifying which (simultaneous) randomized benchmarking pattern is to be performed. The notation `[[i,j],[k]]` means we perform 2 qubits RB on the qubit pair (i, j) while simultaneously performing single qubit RB on qubit k.
  - A list of integers `length_multiplier` specifying by which integer the length vector of the respective index needs to be multiplied. That is, for the pattern `[[i,j],[k]]` the length multiplier `[m1,m2]` multiplies the length vector corresponding to benchmarking pair `[i,j]` with `m1` and that of the singlet `[k]` by `m2`. This step is necessary to compensate for the typical better fidelities of single-qubit operations.
- A list of `basis_gates` specifying which basis gates the Clifford operation will be decomposed into. In this report the single-qubit gates used are `U1,U2,U3` which are parameterized rotations in the computational basis. The two-qubit basis gate used is the `CZ` gate.
- The number of `shots` performed per sequence, i.e the number of measurements done to determine the survival probability of a single RB sequence.

When the experiment is initialized a randomized benchmarking sequence is constructed using Qiskit's **randomized_benchmarking_sequence** under the constraints of the `rb_opts`. This returns a list of Clifford sequences `rb_circs` for each sub-pattern and the appropriately scaled version of the `length_vector` called `xdata`. In order to execute the experiment the following methods are defined.

Methods

The **execute** method performs the RB experiment. First, the Qiskit transpiler is used to translate the circuits `rb_circs` from Clifford gates to the specified basis gates. An example of such a transpiled circuit is shown in Figure 3.3. These gates are passed to the native execute function which runs the simulation using the given noise model.

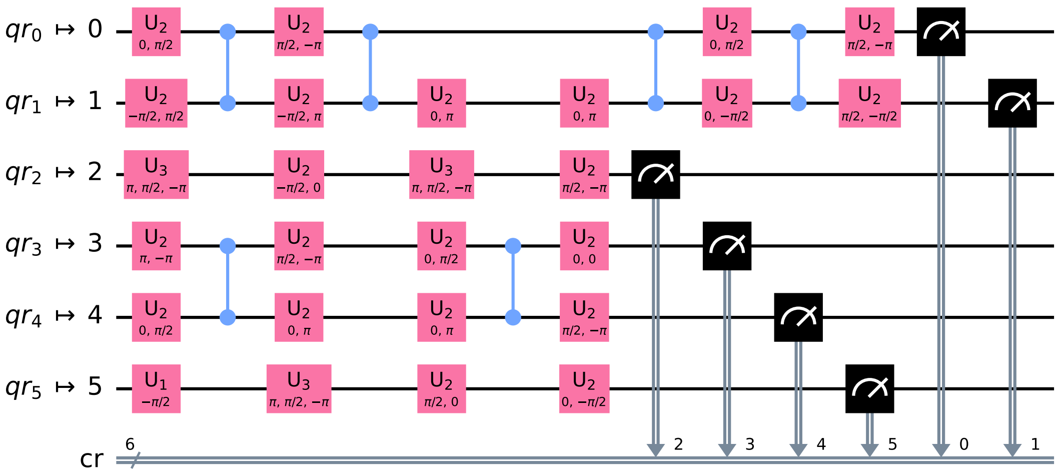

Figure 3.3: Example of a randomized benchmarking sequence on 6 qubits. The displayed sequence is the result of a `[[0,1],[2],[3,4],[5]]` pattern with length multiplier `[1,3,1,3]`. Each of the horizontal black lines represents a qubit register. The gates performed are represented by rectangles in the case of of single qubit gates and with two connected dots for two qubit gates. Measurements are represented by a dial icon.

The **rb_fit** method passes the result of the experiment to an instance of Qiskit's `RBfitter` class which performs the exponential fit for each sub-pattern. The returned fitted decay parameters are then stored in a dictionary according to the index of the qubit (pair) they apply to. The result of the experiment along with the fitted curve can then be plotted using **plot_fit**.

One of the requirements of the novel prediction scheme is knowledge of the amount of single- and two-qubit gates an average Clifford gate is comprised of. There are several hurdles to overcome when attempting to implement a method of to these numbers. Unfortunately, when transpiling the Clifford sequences to basis gate sequences, Qiskit's transpiler does not keep track of the number of underlying gates used per Clifford on average. What is offered is the ad hoc function **gates_per_clifford** which attempts to count the gates per Clifford of a transpiled circuit list. However, this function has the shortcoming that it fails to distinguish between different sub-patterns of the experiment. Since we allow for SWAP operations to occur between neighboring qubits it is unable to keep track of specific sub-experiments. To circumvent this problem we reasonably assume that number of single- and two-qubit gates per Clifford does not depend on which qubit (pair) the sequence is executed on. Additionally, we assume that the effect of SWAP operations has a negligible to non-existent effect on the average number of gates per Clifford. The approach is then to run non-simultaneous RB sequences and use **gates_per_clifford** to do the counting. Preferably these sequences are long (here sequences of length 1000 are used) such that the averaging is sufficiently accurate. We then define **get_gpc2** which uses **gates_per_clifford** to count `N_1_per_1`, `N_1_per_2` and `N_2_per_2`. These are the number of single-qubit basis gates per single-qubit Clifford, the number of single-qubit basis gates per two-qubit Clifford, and the number of two-qubit basis gates per two-qubit Clifford, respectively.

In order to assess the validity and accuracy of prediction of the average n-qubit gate infidelity that will be obtained from the protocol, we must also find the actual average n-qubit gate infidelity. To achieve this we must perform up to 6-qubit randomized benchmarking. This is usually infeasible because the Clifford group cannot be effectively sampled because of its superexponential scaling with the number of qubits. When the sampling is successful, the overhead from transpiling the circuits is also expected to be infeasible to impossible to implement in practice. Conveniently, Qiskit has some pre-defined, pre-transpiled n qubit randomized benchmarking sequences, at least up to and including $n = 6$. It is for this reason that we are able to numerically simulate up to 6-qubit randomized benchmarking. Because a single 6-qubit Clifford consists of many one- and two-qubit basis gates the total sequence lengths blow up quickly. An example of the implementation of a single 6-qubit Clifford is shown in Figure 3.4. Note that only about 1/6th of the entire circuit comprising the single Clifford is shown.

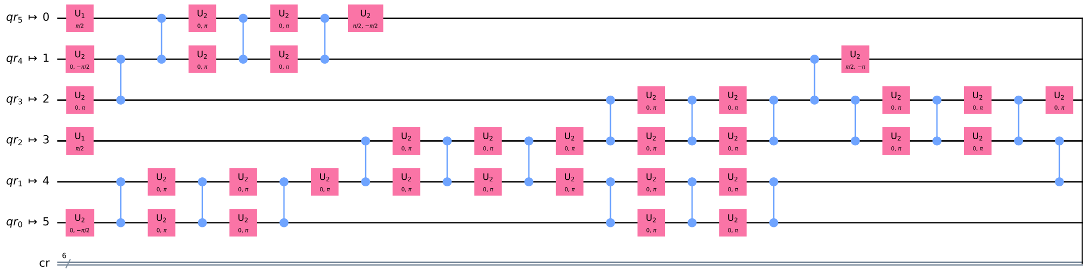

Figure 3.4: Example of a Clifford gate on 6 qubits. Due to the large amount of single- and two-qubit basis gates comprising a single 6-qubit Clifford only about 1/6th of the circuit is shown. Each of the horizontal black lines represents a qubit register. The gates performed are represented by rectangles in the case of single-qubit gates and with two connected dots for two-qubit gates.

To obtain an estimate for the n-qubit fidelity we must measure the fidelity of all single qubits and all pairs of qubits. A complete prediction experiment, therefore, consists of several randomized benchmarking experiments. In the following sections we elaborate on how this can be implemented and exactly which benchmarking patterns are executed. How the results of these experiments are used to make the prediction is also explained.

3.2.2. Complete Benchmarking Experiment

The way a complete benchmarking experiment is handled is through the **FullExperimentNQ** class. Within an instance of this class, we store all comprising **Experiment** instances in an `experiments` list. This allows for the convenient handling of a complete experiment by executing, fitting, and plotting all its experiments at once. This class also contains several methods that handle the prediction, which will be dealt with in the following sections.

Random patterns

When constructing a complete experiment we must choose which underlying experiments to perform. Our only hard constraint is that every single qubit and each pair must be benchmarked once, which is not enough to remove all ambiguity. As the number of qubits n increases, the number of pairs of qubits scales as n^2. We must then construct patterns consisting of these pairs and individual qubits; the singlets. These patterns can be arbitrary and do not necessarily need to work on the full qubit space all at once. However, to catch crosstalk we impose the condition that the pattern should strive for maximum simultaneousness. Evidently, it is not possible to benchmark a singlet and a pair containing this singlet simultaneously. Here we propose a simple algorithm to construct a random benchmarking pattern. We may see all available singlets as vertices of a graph, while the pairs are the corresponding edges. In the case that we benchmark each pair, the aforementioned graph is complete, i.e. each vertex is connected to each other vertex via an edge. A simple method of constructing a viable random pattern works as follows. We start with an empty array to contain our patterns, call it A, and the complete graph G and we make a copy of it, say G_0. From the G_0 we uniformly randomly choose a vertex or edge. Whatever element of the graph is chosen, we remove it from G and G_0 and store the pair or singleton in a pattern array p_0. If the element is a vertex, remove all of its edges from G_0 too. If the element is an edge; delete both its endpoints from G_0, along with all edges connected to either of the endpoints. Continue this process until G_0 is empty, then append p_0 to A. We now discard G_0 and initialize a new graph G_1 as a copy of G. Repeat these steps until G is empty. The result is an array $A = [p_0, ..., p_k]$ for some finite k, with each p_i a pattern array.

This process can alternatively be encoded in an $n \times n$ upper triangle matrix M with ones as entries. Here M has the same role as G and we create a sequence $\{M_i\}$ as before. We now randomly pick nonzero elements of M_i and set all entries of the column and row of the entry and its corresponding transpose entry to zero. We repeat until M is the 0-matrix. The result of both approaches is the same. The latter approach is used and implemented as `get_random_patterns`. This function also takes care of assigning a `length_multiplier` to each pattern.

3.2.3. Predicting fidelity

Under our assumption that the underlying basis gates are depolarising and that we may compose them we find (see Appendix A) in the simple 1D case that

$$\alpha_{i,C} = \alpha_i^{N_1}, \tag{3.5}$$

where $\alpha_{i,C}$ is the fidelity found from single-qubit randomized benchmarking on qubit i, α_i is the average 1-qubit fidelity of qubit i and N_1 is the average number of basis gates per single qubit Clifford. This relation may be inverted to infer the single-qubit basis gate fidelities from the measured benchmarking value

$$\alpha_i = \alpha_{i,C}^{\frac{1}{N_1}}. \tag{3.6}$$

A dictionary containing the single-qubit basis gate fidelities is constructed by the **get_1q_basis_alphas** method of the **FullExperimentNQ** class. It does so by finding the experiment which contains the relevant single qubit benchmark and performing the calculation described by equation 3.6. For the 2 qubit case we find a less trivial expression (see Appendix A)

$$\alpha_{ij,C} = \frac{1}{5}\left(\alpha_i^{N_1} + \alpha_j^{N_1} + 3\alpha_i^{N_1}\alpha_j^{N_1}\right)\alpha_{ij}^{N_2} \tag{3.7}$$

which for $i \neq j$ predicts the two qubit fidelity when benchmarking the pair (i, j) given the respective singlet and pairwise basis gate fidelities. Note that N_1 and N_2 are now the number of one and two qubit basis gates per two qubit Clifford respectively. By inverting this relation we obtain an expression for α_{ij}

$$\alpha_{ij} = \left(\frac{5\alpha_{ij,C}}{\alpha_i^{N_1} + \alpha_j^{N_1} + 3\alpha_i^{N_1}\alpha_j^{N_1}}\right)^{\frac{1}{N_2}}. \tag{3.8}$$

This expression is useful since $\alpha_{ij,C}$ may be found experimentally and α_k may be found from equation 3.6 after measuring $\alpha_{k,C}$. An dictionary containing the two-qubit basis gate fidelity is computed using the **get_2q_basis_alphas** method of the **FullExperimentNQ** class in a similar fashion as for the single-qubit parameters.

Since almost all $n \geq 3$ qubit operations are built out of single- and two-qubit operations, applying Equation 2.37 produces an expression dependent solely on single- and two-qubit basis gate fidelities and the average number of single- and two-qubit basis gates per n-qubit Clifford. The latter of which are again referred to as N_1 and N_2 respectively. What we find is that by only benchmarking singlets and pairs we can calculate a prediction for the result of n-qubit randomized benchmarking, effectively predicting the multi-qubit fidelity. This functionality is implemented as **get_nq_alpha_c**.

3.2.4. Confidence bound propagation

We wish to construct a confidence bound for the prediction. The way this is approached is by using the confidence bounds of the underlying RB experiments. Since our prediction model is a strictly increasing function of the underlying gate fidelities the calculation may be significantly simplified. Along with each fidelity parameter α (i.e. α_i or $\alpha_{i,j}$, depending on the number of qubit being benchmarked) we also store the result $\alpha + \sigma$ and $\alpha - \sigma$, where σ is the standard deviation of the respective fidelity measurement. The whole prediction calculation is then executed using not only the mean α values, but also $\alpha + \sigma$ and $\alpha - \sigma$. The resultant values of these latter two calculations then provide a confidence region around the predicted value.

3.3. Testing approach

We now turn our attention to establishing the performance of the protocol. To do so several numerical experiments are performed on different noise models. Before elaborating on the applied noise model we first provide some of the used values of relevant variables. The values are summarized in Table 3.1.

Table 3.1: Summary of values of relevant randomized benchmarking variables used in the one- and two-qubit numerical experiments. The usage of these variables is explain in section 3.2.1. The length vector and multiplier are chosen in such a way that the full characteristic exponential decay due to twirling is captured given the used underlying gate fidelity.

Variable	Value
`nQ`	3-6
`basis_gates`	[u1, u2, u3, cz]
`shots`	10000
`length_vector`	np.arange(1,401,20)
`length_multiplier`	3 for 1Q-RB, 1 for 2Q-RB
`nseeds`	30, 60, 90, 120, 150

By varying the number of qubits from 3 to 6 we attempt to gain some insight into the degree to which the method is scalable. The chosen sequence length vector ranges from length 1 sequences to length 401 sequences in steps of 20. Together with the specified length multiplier, this ensures that two-qubit subpatterns use the aforementioned sequence length vector while 1 qubit subpatterns range from length 3 to length 1203 in steps of 60. The number of seeds is initially set to 30 but is increased to 150 in steps of 30 to investigate whether the predicted result converges to the expected result from n-qubit RB. For the case of n-qubit simulations, we must adjust the length vector to account for the fact that the decay rate is much higher. We must therefore collect more data points in the low sequence length range, while data points in the high sequence length range lose value because the decay has already occurred at those lengths.

3.3.1. Noise model implementation

For testing purposes, it is interesting to investigate the effectivity of the protocol under multiple noise models. We simulate here two noise models which are each of practical interest for different reasons. A depolarizing channel is the simplest noise model to implement and relates directly to the randomized benchmarking protocol and the underlying mathematics of the prediction scheme. For this reason, it is useful in the process of designing the protocol. Dephasing is one of the most common types of noise encountered in actual spin qubit setups. Additionally, while distinct, dephasing and depolarising errors are mathematically quite similar. This makes dephasing noise ideal as a first non-depolarising testing noise model.

Depolarizing

One part of the protocol deals with inferring the 1 and 2 qubit basis gate fidelities from 1 and 2 qubit randomized benchmarking. In the case of depolarizing noise, this fidelity is synonymous with the complement of the depolarizing probability. We should therefore be able to check whether this is indeed what we find. Another reason that also makes intuitive sense is that local depolarizing channels should indeed twirl to a global depolarizing channel. The depolarizing channels are simple to implement in Qiskit using its built-in `depolarizing_error` function which requires a depolarizing probability p and the number of qubits on which the error acts. For the single-qubit depolarizing probability we choose $p_{1Q} = 0.002$. For the two-qubit version we choose $p_{2Q} = 0.01$

Dephasing

Dephasing noise is a common noise type encored in the studied device. The dephasing error on single qubits is implemented as a `kraus_error` which given a set of Kraus matrices simulates the noise channel defined by these matrices. The decomposition given in equation equation 2.16 is used with $p = p_{1Q} = 0.002$.

For the two-qubit case, we only consider the simple case of simultaneous dephasing of both qubits. This error is implemented using Qiskit's `pauli_error`. With probability p_{2Q} a ZZ is applied and with probability $1 - p_{2Q}$ the identity is applied. For the two-qubit dephasing probability we again choose $p_{2Q} = 0.01$

3.4. Numerical simulations

In this section the results of the various numerical simulations are presented according to the applied noise model. We begin with the depolarization model and continue with the more realistic dephasing model.

3.4.1. Depolarising noise

Table 3.2 shows the n-qubit simulated fidelity versus the predicted result from single and two-qubit fidelities for 30 seeds. These simulations have been performed for 3, 4, 5, and 6 qubits. Tables 3.3 and 3.4 contain the results for the 3- and 4-qubit cases, respectively, with varying number of seeds.

An interesting observation is that the confidence bound does not appear to scale significantly with the performed number of seeds. More interestingly, the variance among the results for different numbers of seeds does not seem to be captured by the confidence bounds. A possible explanation is that our choice for Clifford lengths is not sufficient to get tight bounds. To see what the result of a simultaneous RB simulation looks like, an example of a fit `[[0],[1,2]]` pattern simulation on 3 qubits has been included in Figure 3.5. The result of benchmarking using 3-qubit Clifford gates is presented in Figure 3.6.

Table 3.2: Numerically simulated results and corresponding predictions for the average n-qubit gate fidelity in the case of depolarizing noise. The simulation has been performed for $3-6$ qubits. The confidence bound of the n-qubit result is given by the fit using the least square error method. The confidence bound of the prediction is estimated by propagated confidence bounds of the single- and two-qubit average gate fidelities.

number of qubits	nQ-result	Prediction
3	0.915±0.0019	0.915±0.0025
4	0.771±0.0045	0.755±0.0017
5	0.51258±0.00615	0.50319±0.00416
6	0.32054±0.00752	0.32067±0.00416

Table 3.3: Numerically simulated results and corresponding predictions for the average 3-qubit gate fidelity in the case of depolarizing noise. The number of seeds has been varied from 30 to 150 in steps of 30. The confidence bound of the 3-qubit result is given by the fit using the least square error method. The confidence bound of the prediction is estimated by propagated confidence bounds of the single- and two-qubit average gate fidelities.

number of seeds	3Q-result	Prediction
30	0.904±0.0074	0.905±0.0078
60	0.90533±0.00033	0.90561±0.00033
90	0.90570±0.00031	0.90608±0.00019
120	0.90473±0.00028	0.90512±0.00022
150	0.90601±0.00021	0.90650±0.00018

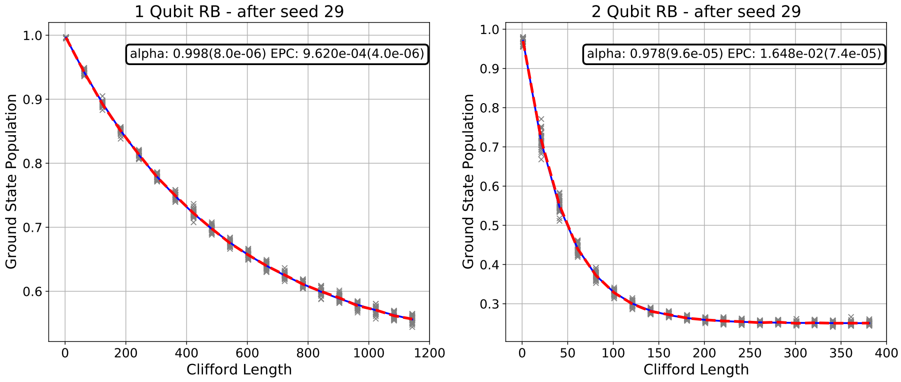

Figure 3.5: Exponential fit to the result of simultaneous randomized benchmarking using a `[[0],[1,2]]` *pattern with 30 seeds. At a fixed Clifford length, each cross represents the result of measurement of a single RB sequences (i.e. a single seed). The length vector and multiplier are specified in Table 3.1. The applied noise model is a depolarizing channel for 1- and 2-qubit basis gates.*

Table 3.4: Numerically simulated results and corresponding predictions for the average 4-qubit gate fidelity in the case of depolarizing noise. The number of seeds has been varied from 30 to 120 in steps of 30. The confidence bound of the 4-qubit result is given by the fit using the least square error method. The confidence bound of the prediction is estimated by propagated confidence bounds of the single- and two-qubit average gate fidelities.

number of seeds	4Q-result	Prediction
30	0.771±0.0045	0.755±0.0017
60	0.70897±0.00161	0.69623±0.00345
90	0.70498±0.00125	0.71028±0.00059
120	0.70973±0.00114	0.69896±0.00306

Table 3.5: Numerically simulated results and corresponding predictions for the average n-qubit gate fidelity in the case of dephasing noise. The simulation has been performed for $3-6$ qubits. The confidence bound of the n-qubit result is given by the fit using the least square error method. The confidence bound of the prediction is estimated by propagated confidence bounds of the single- and two-qubit average gate fidelities.

number of qubits	nQ-result	Prediction
3	0.906±0.0011	0.908±0.0008
4	0.715±0.0041	0.713±0.0015
5	0.524±0.0057	0.528±0.0020
6	0.344±0.015	0.346±0.0020

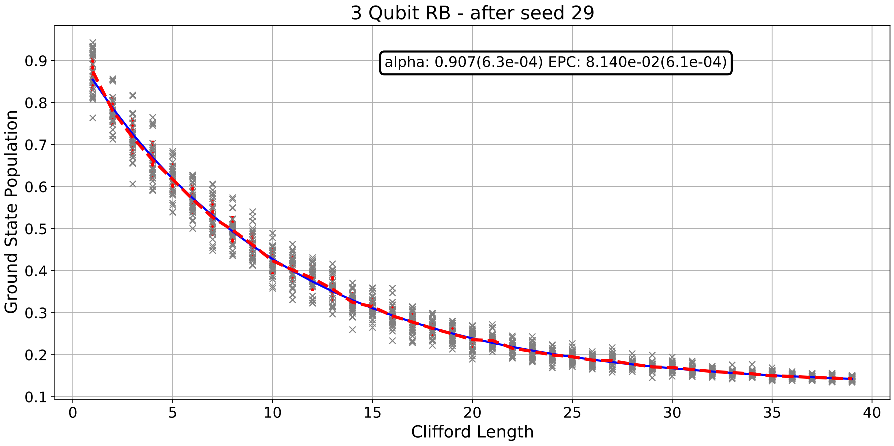

Figure 3.6: Exponential fit to the result of randomized benchmarking using a `[[0,1,2]]` pattern with 30 seeds. At a fixed Clifford length, each cross represents the result of measurement of a single RB sequences (i.e. a single seed). The length vector ranges from length 1 to 39 in steps of 1. The applied noise model is a depolarizing channel for 1- and 2-qubit basis gates.

3.4.2. Dephasing noise

In Table 3.5 the n-qubit simulated fidelity versus the predicted result from single and two-qubit fidelities is shown for 30 seeds for a dephasing noise model. These simulations have been performed for 3, 4, 5, and 6 qubits. Tables 3.6 and 3.7 contain the results for the 3- and 4-qubit case respectively, with varying number of seeds.

The same comments about the confidence bounds in the case of depolarizing noise apply to the dephasing case as well. The scaling of the confidence bound with the number of seeds is insignificant. Again, the confidence bounds of the results for different numbers of seeds do not all overlap. In Figures 3.7 and 3.8 we have plotted two distinct cases of simultaneous two-qubit RB, both performed as part of the same 4-qubit prediction simulation. In Figure 3.7 the pair of qubits are nearest neighbours while in Figure 3.8 non-neighbouring qubits are chosen.

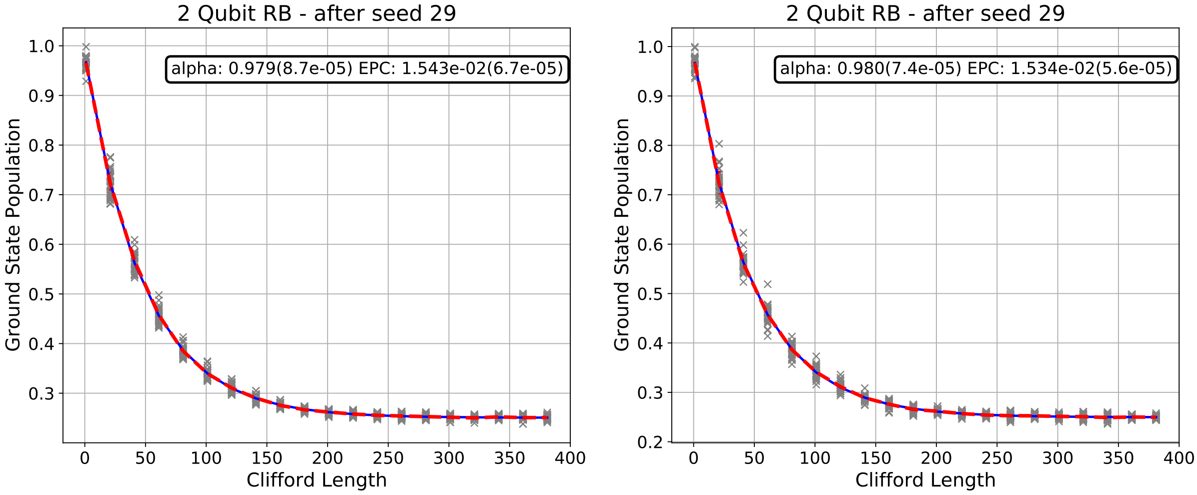

Figure 3.7: Exponential fit to the result of simultaneous randomized benchmarking using a `[[0,1],[2,3]]` pattern with 30 seeds. At a fixed Clifford length, each cross represents the result of measurement of a single RB sequences (i.e. a single seed). The length vector and multiplier are specified in Table 3.1. The applied noise model is a dephasing channel for 1- and 2-qubit basis gates.

Table 3.6: Numerically simulated results and corresponding predictions for the average 3-qubit gate fidelity in the case of dephasing noise. The number of seeds has been varied from 30 to 150 in steps of 30. The confidence bound of the n-qubit result is given by the fit using the least square error method. The confidence bound of the prediction is estimated by propagated confidence bounds of the single- and two-qubit average gate fidelities.

number of seeds	3Q-result	Prediction
30	0.906±0.0011	0.908±0.0008
60	0.90920±0.00027	0.90911±0.00029
90	0.90843±0.00035	0.90865±0.00018
120	0.90785±0.00037	0.90863±0.00022
150	0.90793±0.00022	0.90897±0.00017

Table 3.7: Numerically simulated results and corresponding predictions for the average 4-qubit gate fidelity in the case of dephasing noise. The number of seeds has been varied from 30 to 120 in steps of 30. The confidence bound of the 4-qubit result is given by the fit using the least square error method. The confidence bound of the prediction is estimated by propagated confidence bounds of the single- and two-qubit average gate fidelities.

number of seeds	4Q-result	Prediction
30	0.715±0.0041	0.713±0.0015
60	0.71173±0.00190	0.71173±0.00190
90	0.71491±0.00130	0.71708±0.00045
120	0.71089±0.00145	0.69517±0.00398

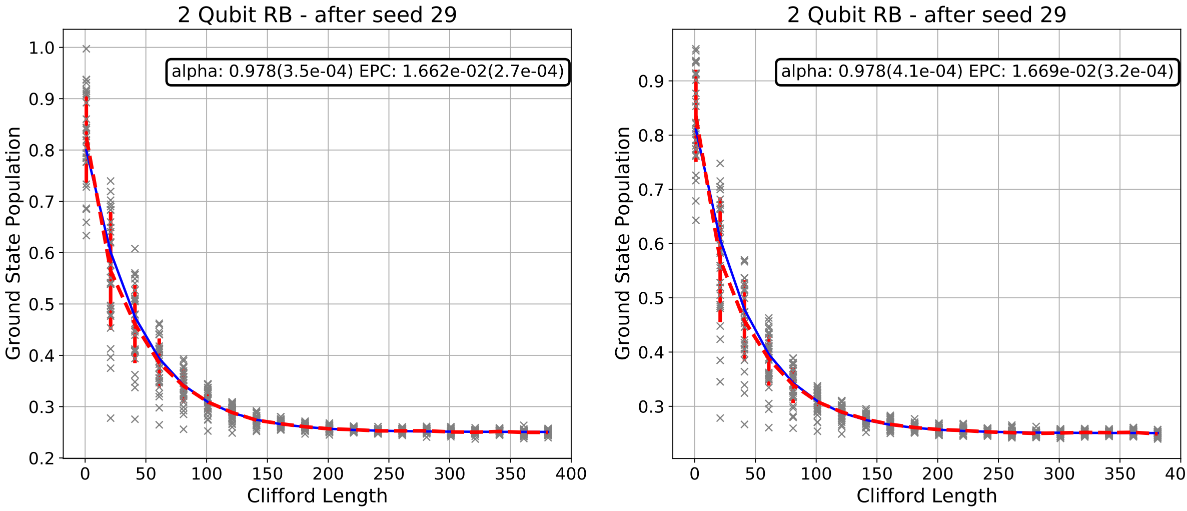

Figure 3.8: Exponential fit to the result of simultaneous randomized benchmarking using a `[[0,4],[1,2]]` pattern with 30 seeds. At a fixed Clifford length, each cross represents the result of measurement of a single RB sequences (i.e. a single seed). The length vector and multiplier are specified in Table 3.1. The applied noise model is a dephasing channel for 1- and 2-qubit basis gates.

It is apparent that the spread in Figure 3.8 is much larger than that in Figure 3.7. This reflects on individual confidence bounds of the benchmarked fidelities as shown in the figures: the difference is around one order of magnitude. This difference in variance is caused by the way in which Qiskit implements two-qubit RB on non-neighboring qubits. The number of SWAP operations required to perform a certain sequence appears to depend significantly on the exact sequence that is performed. The SWAP gates are constructed from noisy CZ gates. The variance thus depends strongly on the number of SWAP operations.

4 | Discussion and Further Ideas

The results following from the numerical simulations are promising at first glance. From Table 3.2 it can be argued that the predicted results give an indication of the value of the true (measured) fidelity. However, we must remark that, at least in the case of 30 seeds, the confidence bounds do in general not overlap. The confidence bounds are of the same order of magnitude, suggesting that under the applied parameters both benchmarking procedures theoretically benchmark to the same statistical level of precision. With the limited data available it is infeasible to draw strong conclusions regarding the accuracy of the result.

Because we are considering a noise model that satisfies all assumptions made in the derivation of the prediction protocol, we expect that the result of the prediction should converge to the result of n-qubit benchmark. From Table 3.3 it is apparent that this expected behavior is not observed. A possible explanation for this is that Qiskit might not sample sufficiently from the Clifford group for $n \geq 3$. It is computationally demanding to construct higher dimensional, properly sampled Clifford sequences. The solution provided by Qiskit might not be sufficiently accurate to obtain the expected result.

An important observation from Table 3.3 is that the scaling of the confidence bounds is certainly not linear. Even though there is limited data we may infer that the confidence bounds shrink significantly in the step from 30 to 60 seeds. Increasing the number of seeds even further does not appear to significantly alter the size of the confidence bounds. These claims are corroborated by the results found for the dephasing case in Table 3.6.

It is possible to construct an upper bound on the size of the confidence bounds by the procedure outlined in [23]. However, the confidence bounds found using our approach are far below the theoretical upper bound. This makes sense considering that our noise model is locally depolarizing and non-gate dependent. Additionally, the number of seeds and the sequence lengths performed are sufficient to capture the full exponential decay. The fact that the confidence bound is tight is therefore not surprising. What is surprising is that the variance among the benchmarking results for different numbers of seeds is inconsistent with these thigh confidence bounds. This last point requires further investigation.

For the case of dephasing noise, the analysis is similar to what has been outlined above. The one exception is that in this case, the underlying noise is no longer depolarizing. Ideally, we would like to make conclusions about the accuracy of the prediction scheme in the depolarizing case. We are also interested in how the accuracy compares to the depolarizing case. However, the results for the dephasing case show the same features in terms of convergence and confidence bounds. For this reason, it is difficult to make conclusions about the effects of a non-depolarizing noise channel based on these results. It is expected that the protocol behaves similarly to the depolarizing case in terms of accuracy because the two noise models are mathematically quite similar. However, the observed noise due to implementation problems is so large that it is infeasible to assess whether a discrepancy between these two cases exists.

In order to assess the effect of the applied noise model, we propose a worst-case scenario to establish whether the prediction model breaks down for certain (non-time-dependent) error types. Because unitary noise is highly non-commutative with the applied Cliffords we choose some unitary noise model as a final test case. We propose the following noise model

- 1 qubit: an overrotation of $\phi/10$ rad on each operation, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. This is used because it the real number that is most difficult to represent as fraction.
- 2 qubits: a rotation of $\theta = 0.01$ rad over the axis described by the generator M, i.e. $e^{-i\theta M}$.

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4.1}$$

One problem with implementing this type of noise model is that the spread between the seeds becomes very large. Unitary noise only affects the fidelity in second order [24]. The result is the the decay is very hard to observe on its own as can be seen from figure 4.1.

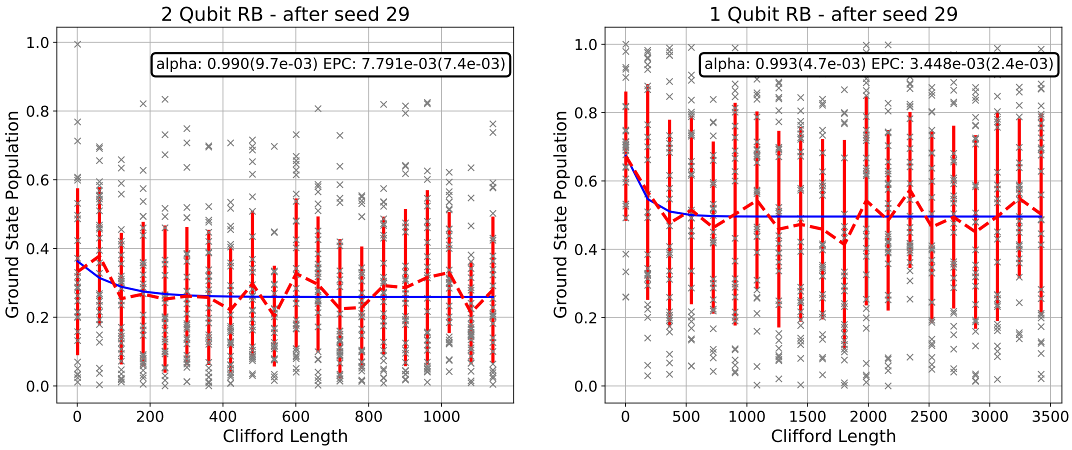

Figure 4.1: Exponential fit to the result of simultaneous randomized benchmarking using a `[[0,1],[2]]` pattern with 30 seeds. At a fixed Clifford length, each cross represents the result of measurement of a single RB sequences (i.e. a single seed). The applied noise model is unitary for 1- and 2-qubit basis gates.

With the current implementation, the protocol is however not able to completely execute. Running a high number of seeds also very computationally intensive. As a consequence of the large spread due to the unitary gates, we expect larger confidence bounds for the predicted and measured fidelity. With enough computational power, it may be possible to reveal whether to protocol retains its predictive power under unitary errors.

A future study must also reveal how well the usage of random patterns captures crosstalk effects. To do so a noise model that encapsulates these effects must be created. It would then be interesting to see whether the prediction is reasonable and whether the predicted value depends on the chosen pattern. Since in a real device we do not know which crosstalk effects are most prominent, choosing the benchmarking pattern is difficult. The protocol might therefore not be suited for use without gaining this information from some other protocol first.

Because we perform a random pattern, the chosen pattern might be different between multiple prediction benchmarking experiments. Some patterns might require more SWAP operations, which as we have seen in in Figures 3.7 and 3.8 can have an effect on the underlying benchmarking results. The assumption that the difference between patterns in terms of the number of SWAP operations (and therefore the number of underlying two-qubit gates) is small, might not be valid. A consequence is that the benchmark is potentially imprecise, in the sense that performing the benchmarking protocol on the same device twice may yield different results due to a different pattern being performed. An improved version of the prediction protocol might choose the benchmarking pattern more intelligently, for example under the constraint of minimum applied SWAP operations. This requires more interference with, and a detailed understanding of, the protocols used in Qiskit and might warrant the use of a different compilation package altogether.

5 | Conclusions

We have proposed a protocol that predicts the multi-qubit average gate fidelity by performing simultaneous randomized benchmarking on subspaces, overcoming the scalability issues of standard randomized benchmarking. Such a protocol fulfills the need for an efficient benchmarking protocol for assessing the fidelity of multi-qubit devices. The protocol only requires benchmarking of single qubits and pairs of qubits to predict the outcome of a multi-qubit randomized benchmarking experiment. Under the assumption that the fidelity of the underlying gates can be modeled by a depolarizing parameter, it is possible to calculate the average depolarization parameter associated with the underlying single- and two-qubit basis gates using the result of the subsystem benchmarks. With knowledge of the number of single- and two-qubit basis gates making up the average n-qubit Clifford it is possible to compose the underlying maps in order to predict the result of n-qubit randomized benchmarking.

The protocol must benchmark every single qubit and each pair of qubits at least once. The amount of possible benchmarking patterns scales quadratically with the number of qubits. We choose a random pattern under the objective that operations are preferably performed simultaneously in order to catch cross-talk effects and reduce overhead.

The design, implementation, and validation through simulation of the protocol were facilitated by IBM's Qiskit tool for quantum computing. The two main noise models that have been tested are depolarizing and dephasing noise. For both types of errors, the prediction results are close to the results obtained from performing the multi-qubit randomized benchmarking experiment as displayed in Tables 3.2 and 3.3. A confidence bound for the prediction is provided using the variance obtained directly from the fits of the RB experiments performed in the single- and two-qubit subspaces.

With the aim of verifying that the prediction converges to the desired result, the protocol has been simulated using a varying number of seeds. While the confidence bound does tighten with an increasing number of seeds, we were not able to draw the conclusion that the accuracy of the result is completely captured by the confidence bound. The spread between the benchmarking values resulting from experiments with different total seed numbers is larger than the calculated error bounds. Several reasons as to why this could be the case include shortcomings of the Qiskit tool, insufficient data points, or the nature of the prediction scheme; this requires further investigation and perhaps compilation using a different simulation package. This unexpected behavior also prevents us from drawing statistically convincing conclusions about the effectivity of the protocol under different types of noise models.

We have additionally proposed a further worst-case scenario test based on a unitary noise model since this is theoretically the most unfavorable error type for the prediction scheme. We are positive that once the mentioned issues are solved and statistically verified that our protocol can be used to accurately benchmark multi-qubit devices efficiently.

References

[1] Mark Friesen, Paul Rugheimer, Donald E. Savage, Max G. Lagally, Daniel W. van der Weide, Robert Joynt, and Mark A. Eriksson. Practical design and simulation of silicon-based quantum-dot qubits. *Phys. Rev. B*, 67:121301, Mar 2003.

[2] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[3] Todd A. Brun. Quantum error correction, 2019.

[4] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I.-J. Wang, Simon Gustavsson, and William D. Oliver. Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, 11(1):369–395, Mar 2020.

[5] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2):021314, Jun 2019.

[6] Viktoria Yurgens, Josh A. Zuber, Sigurd Flågan, Marta De Luca, Brendan J. Shields, Ilaria Zardo, Patrick Maletinsky, Richard J. Warburton, and Tomasz Jakubczyk. Low charge-noise nitrogen-vacancy centers in diamond created using laser writing with a solid-immersion lens, 2021.

[7] Y. Xu, F. K. Unseld, A. Corna, A. M.J. Zwerver, A. Sammak, D. Brousse, N. Samkharadze, S. V. Amitonov, M. Veldhorst, G. Scappucci, R. Ishihara, and L. M.K. Vandersypen. On-chip integration of si/sige-based quantum dots and switched-capacitor circuits. *Applied Physics Letters*, 117(14), 2020. Green Open Access added to TU Delft Institutional Repository 'You share, we take care!' – Taverne project https://www.openaccess.nl/en/you-share-we-take-care Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

[8] Seth T. Merkel, Jay M. Gambetta, John A. Smolin, Stefano Poletto, Antonio D. Córcoles, Blake R. Johnson, Colm A. Ryan, and Matthias Steffen. Self-consistent quantum process tomography. *Physical Review A*, 87(6), Jun 2013.

[9] Joseph Emerson, Robert Alicki, and Karol Życzkowski. Scalable noise estimation with random unitary operators. *Journal of Optics B: Quantum and Semiclassical Optics*, 7(10):S347–S352, Sep 2005.

[10] X. Xue, T. F. Watson, J. Helsen, D. R. Ward, D. E. Savage, M. G. Lagally, S. N. Coppersmith, M. A. Eriksson, S. Wehner, and L. M. K. Vandersypen. Benchmarking gate fidelities in a $\mathrm{Si/SiGe}$ two-qubit device. *Phys. Rev. X*, 9:021011, Apr 2019.

[11] Jonas Helsen, Xiao Xue, Lieven M. K. Vandersypen, and Stephanie Wehner. A new class of efficient randomized benchmarking protocols, 2019.

[12] Daniel Greenbaum. Introduction to quantum gate set tomography, 2015.

[13] Easwar Magesan, J. M. Gambetta, and Joseph Emerson. Scalable and robust randomized benchmarking of quantum processes. *Physical Review Letters*, 106(18), May 2011.

[14] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5), Nov 2004.

[15] Easwar Magesan, Jay M. Gambetta, and Joseph Emerson. Characterizing quantum gates via randomized benchmarking. *Phys. Rev. A*, 85:042311, Apr 2012.

[16] Jay M. Gambetta, A. D. Córcoles, S. T. Merkel, B. R. Johnson, John A. Smolin, Jerry M. Chow, Colm A. Ryan, Chad Rigetti, S. Poletto, Thomas A. Ohki, Mark B. Ketchen, and M. Steffen. Characterization of addressability by simultaneous randomized benchmarking. *Phys. Rev. Lett.*, 109:240504, Dec 2012.

[17] David C. McKay, Sarah Sheldon, John A. Smolin, Jerry M. Chow, and Jay M. Gambetta. Three-qubit randomized benchmarking. *Physical Review Letters*, 122(20), May 2019.

[18] DM Zajac, TM Hazard, Xiao Mi, E Nielsen, and Jason R Petta. Scalable gate architecture for a one-dimensional array of semiconductor spin qubits. *Physical Review Applied*, 6(5):054013, 2016.

[19] AMJ Zwerver, T Krähenmann, TF Watson, L Lampert, HC George, R Pillarisetty, SA Bojarski, P Amin, SV Amitonov, JM Boter, et al. Qubits made by advanced semiconductor manufacturing. *arXiv preprint arXiv:2101.12650*, 2021.

[20] Menno Veldhorst, CH Yang, JCC Hwang, W Huang, JP Dehollain, JT Muhonen, S Simmons, A Laucht, FE Hudson, Kohei M Itoh, et al. A two-qubit logic gate in silicon. *Nature*, 526(7573):410–414, 2015.

[21] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, and M. Veldhorst. Interfacing spin qubits in quantum dots and donors–hot, dense, and coherent. *npj Quantum Inf.*, 3(1):34, 2017.

[22] Héctor Abraham et al. Qiskit: An open-source framework for quantum computing, 2019.

[23] Joel J Wallman and Steven T Flammia. Randomized benchmarking with confidence. *New Journal of Physics*, 16(10):103032, Oct 2014.

[24] Robin Blume-Kohout, Marcus P. da Silva, Erik Nielsen, Timothy Proctor, Kenneth Rudinger, Mohan Sarovar, and Kevin Young. A taxonomy of small markovian errors, 2021.

Appendix

A: Derivation of prediction formulas

For the one-dimensional case we have as the PTM representation for a depolarizing channel with parameter the α_i

$$R_\Lambda = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_0 & 0 & 0 \\ 0 & 0 & \alpha_0 & 0 \\ 0 & 0 & 0 & \alpha_0 \end{bmatrix} \tag{A1}$$

If we know that a single qubit Clifford consists on average of N_1 single qubit basis gates we can find the error channel corresponding to the average single qubit Clifford by composing the underlying depolarizing channel N_1 times yielding

$$R_\Lambda^{N_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha_0^{N_1} & 0 & 0 \\ 0 & 0 & \alpha_0^{N_1} & 0 \\ 0 & 0 & 0 & \alpha_0^{N_1} \end{bmatrix}. \tag{A2}$$

The result of a twirl according to equation is then 2.37

$$\frac{\text{Tr}(R_\Lambda^{N_1}) - 1}{4 - 1} = \frac{3\alpha_0^{N_1}}{3} = \alpha_0^{N_1} \tag{A3}$$

as expected. For the two qubit case the approach remains the same but we now have to deal with non-trivial subspaces. For the average depolarizing channel with parameter α_0 acting on qubit 0 we obtain

$$R_{\Lambda,0} = \text{diag}(1, 1, 1, 1, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0, \alpha_0). \tag{A4}$$

Similarly, for the average depolarizing channel with parameter α_1 acting on qubit 1 we obtain

$$R_{\Lambda,1} = \text{diag}(1, \alpha_1, \alpha_1, \alpha_1, 1, \alpha_1, \alpha_1, \alpha_1, 1, \alpha_1, \alpha_1, \alpha_1, 1, \alpha_1, \alpha_1, \alpha_1). \tag{A5}$$

On the space of two qubits a depolarizing channel channel with parameter α_{01} is described by

$$R_{\Lambda,01} = \text{diag}(1, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}, \alpha_{01}). \tag{A6}$$

Composing these maps into the average Clifford yields

$$R_{\Lambda,C} = R_{\Lambda,0}^{N_1} R_{\Lambda,1}^{N_1} R_{\Lambda,01}^{N_2}, \tag{A7}$$

where N_1 and N_2 are the average number of single- and two-qubit gates per two-qubit Clifford. Applying equation 2.37 we find

$$\frac{\text{Tr}(R_{\Lambda,C}) - 1}{16 - 1} = \frac{3\alpha_0^{N_1} + 3\alpha_1^{N_1} + 9\alpha_0^{N_1}\alpha_1^{N_1}}{15}\alpha_{01}^{N_2}, \tag{A8}$$

which reduces to equation 3.7 as expected.

B: Code

The implementation of the protocol as well as the simulations is available at: `https://github.com/dveldhuiz/BEP-RB`