Particle Swarm Optimisation to Infer Quantum Computer Circuit Errors

Chris Wise, Matt Woolley
School of Engineering and Information Technology
University of New South Wales
Canberra, Australia
c.wise@student.unsw.edu.au, m.woolley@unsw.edu.au

***Abstract*—To go beyond the current noisy intermediate-scale quantum era, we need to minimise the effects of noise in quantum computers. This paper applies machine learning (ML) techniques to address this noise in quantum computers. Specifically, we inferred quantum computer circuit error parameters using particle swarm optimisation (PSO). Using the PSO algorithm, we optimised a circuit error model. We reduced the mean squared error (MSE) between the model's simulated data and experimental data by a magnitude of ten compared to an error-free model. We then used the PSO algorithm to find control parameters that mitigate the effects of circuit errors. The PSO-optimised control parameters reduced the MSE between measured noisy data and ideal error-free data by 62% when compared to data from circuits with no control parameters implemented.**

***Index Terms*—Quantum Computing · Machine Learning · Particle Swarm Optimisation**

I. Introduction

Future quantum computers will efficiently solve problems currently intractable to digital (classical) computers. Examples of such problems include prime factorisation of large numbers [19], searching of unordered lists [3], and simulations of many-body quantum systems [13]. Quantum computers achieve their superior performance using qubits. Quantum bits or qubits are built from the quantum property of an object. Qubits enable superior computing performance by exploiting the ability of a quantum property to exist in multiple states simultaneously, a superposition of states [15].

However, despite having efficient specialised quantum computer algorithms and working quantum computers, these problems above remain intractable as we are in a noisy intermediate-scale quantum (NISQ) era. To move beyond NISQ, we need to minimise, account for, and mitigate noise [16].

Noise in quantum computers is a consequence of decoherence resulting from unwanted environmental quantum interactions and state preparation and measurement (SPAM) errors resulting from miscalibrated experimental apparatus [2].

Unfortunately, quantum computer circuit errors interact in a highly complex non-linear manner and often cannot be isolated [17], making complete study and diagnostics of circuit errors challenging [10]. This complexity of circuit errors has motivated our use of machine learning (ML) techniques to study them.

Specifically, we inferred circuit errors using a parameterised model to simulate the measurement statistics of a quantum computer. Our work used the particle swarm optimisation (PSO) algorithm [6] to infer these circuit errors. The PSO algorithm was used to find model parameters that minimised the mean squared error (MSE) between the model's simulated data and measured data from a live quantum computer. We extended our PSO techniques to optimise control parameters that mitigate the effects of circuit error in quantum computer simulations.

Developing techniques to understand and diagnose circuit errors is essential, as these techniques can be applied to quantum computers to allow efficient and effective system diagnostics. Further, isolating the effects of SPAM errors from unwanted environmental interaction is necessary to allow correct mitigation of SPAM errors at a software level.

Following this introduction, Section II provides background on qubits, circuit errors, and the PSO algorithm. Experiment methodologies are explained in Section III, with Section IV detailing experiment results. Experiment and result validity is discussed in Section V, and concluding remarks follow in Section VI. The paper finishes with Section VII, which provides suggestions for future circuit error studies.

II. Background

A. Qubits and the Bloch Sphere

Qubits are constructed using a quantum property of an object, with states of that quantum property representing 0 and 1. Being a quantum property, one can manipulate this qubit into a superposition of states.

A qubit state, $|\psi\rangle$, can be represented as a vector in a vector space, with an associated basis in which the vector can be expanded. In quantum computing, the computational basis is often chosen and is analogous to 0 and 1 in classical computing, with 0 and 1 represented as $|0\rangle$ and $|1\rangle$, respectively:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The qubit's state is described as a linear combination of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$.

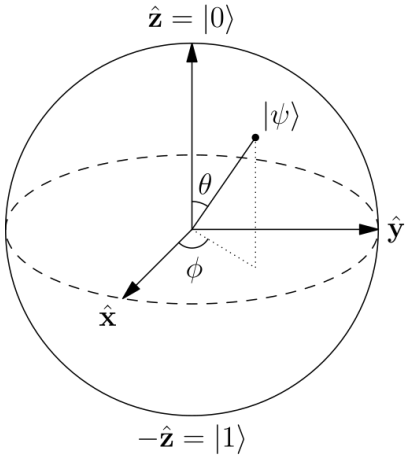

Fig. 1. Illustration of the Bloch sphere [7] which can describe qubit states, where conventionally, north and south poles are the computation basis.

Qubits are often described using the Bloch sphere, a unit 2-sphere where conventionally, north and south poles are the computation basis. By using the Bloch sphere, one can describe qubit states using trigonometry:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle$$

where $|\alpha|^2 + |\beta|^2 = 1$, $0 \leq \theta \leq \pi$ and $0 \leq \theta < 2\pi$. The ϕ value describes the qubit's phase, with this phase having no effect when measuring against the computation basis [7]. An illustration of the Bloch sphere can be seen in Figure 1.

Computation is represented by the qubit "rotating" to another position on the Bloch sphere. These rotations are described in a generic gate matrix [1]:

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\lambda+\phi)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

The state of a qubit is measured by projecting its state onto a measurement basis. Though any arbitrary basis is possible, we will focus on the zero basis vector as it is the most relevant basis vector to this paper. The zero basis vector can be described using a density matrix:

$$|0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

When projecting onto the zero basis vector, for a given qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the probability of measuring 0 is:

$$|\alpha|^2 = \cos^2\left(\frac{\theta}{2}\right)$$

B. Circuit Errors

Coherent errors result from imprecise or miscalibrated experiment apparatus, and incoherent errors result from unwanted environmental interactions, with incoherent errors being more challenging to predict than coherent errors [5].

Examples of coherent errors include SPAM errors. A preparation error results in the qubit being prepared in a state different to the desired state. Preparation errors can be represented as under- or over-rotation of the qubit [9]. Measurement errors result in projecting a qubit onto a basis vector different from what is expected, affecting measurement statistics [9]. Measurement errors are represented by a basis vector being rotated away relative to the expected basis vector.

Examples of incoherent errors include unwanted environmental interactions. When incoherent errors occur, information is lost to the environment [5], [8]. Incoherent errors are modelled by a qubit's state description moving from the Bloch sphere's surface to inside the sphere. The qubit is then represented as a statistical mixture of states [15]. Incoherent errors can be represented by Kraus operators acting on the qubit's state. Kraus operators are formed as the product of an operator specifying the evolution of the qubit state and a scalar related to the probability of the qubit undergoing that state evolution [12]. For example, two such Kraus operators model a bit-flip error:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$A_0 = \sqrt{1-p}I \quad A_1 = \sqrt{p}X$$

where $0 \leq p \leq 1$, p is the probability of a bit-flip error occurring, X describes the evolution of a qubit bit-flip error, and I describes the evolution of qubit maintaining its current state.

C. Particle Swarm Optimisation (PSO)

The PSO algorithm is a global population-based stochastic optimisation algorithm used to optimise solutions where the derivative of an objective function is difficult or unclear to calculate [11]. The algorithm starts with a population of particles that are randomly distributed in a search space. Then to find an optimal solution, each particle will move towards its personal and the swarm's best-known position. Particles start by recording the current direction they are heading in, the direction of the particle's best personal solution, and the direction of the global best solution the swarm has found [6]. For each iteration, particles move in some combination of these directions. At the end of each iteration, the swarm updates its global best, and particles update their personal best. Additionally, such that particles move stochastically, particles move in randomised amounts to the global and their personal bests, and the distances particles move each iteration decreases to efficiently converge particles to the optimal solution [6].

The i^{th} particle's position at time t is describe as an n-dimensional vector [11]:

$$X_i^t = [x_i^t, y_i^t, z_i^t, ...]$$

For the i^{th} particle at time t, its velocity is V_i^t, the particle's personal best recorded position is P_i^t and the global best

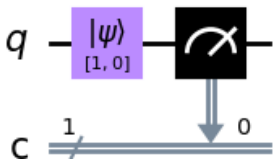

Fig. 2. Example of circuit used for experiments with $\theta = 0$ and $\epsilon = 0$

recorded position is G^t. One can then calculate the velocity and subsequent position at time $t+1$ [11]:

$$V_i^{t+1} = wV_i^t + c_1 r_1 (P_i^t - X_i^t) + c_2 r_2 (G^t - X_i^t)$$
$$X_i^{t+1} = X_i^t + V_i^{t+1}$$

where $\{w, c_1, c_2\} \in \mathbb{R}$ and $0 \leq \{r_1, r_2\} \leq 1$. So that particles move smaller distances when approaching function minima, the parameter w decreases linearly as a function of iterations. Further, c_1 is the cognitive parameter and controls how much particles move towards their personal best, and c_2 is the social parameter and controls how much particles move towards the global best [18]. The values r_1 and r_2 are randomised for each iteration to implement stochastic particle movements [6], [11].

III. Methodology

The Python3 code used for all experiments is available here: https://github.com/ChrisWise07/ZEIT3191-ML-Quantum-Computing

A. Collecting Error Data

The Qiskit library [1] and corresponding IBMQ quantum computers were used to gather measurement statistics in the presence of circuit errors. Specifically, IBMQ Quito was used. All circuits were executed 20,000 times, the limit set by IBMQ. All circuits initialised a qubit in a given state and then measured against the computational basis. An illustration of this circuit can be seen in Figure 2.

For measurement statistics, we estimated the probability of measuring zero in the presence of circuit errors ($\tilde{p}$) as the number of zeros measured (n_0) divided by the total number of executions (n). That is:

$$\hat{\tilde{p}}(0 \mid \theta \wedge \phi) = \frac{(n_0 \mid \theta \wedge \phi)}{n}$$

Initial experiments to acquire measurement statistics used 100 unique circuits. One hundred circuits were executed as IBMQ limits execution jobs to a list of 100 circuits. Circuits were created by initialising qubits with a θ value from a distribution of 10 evenly spaced values across the interval $0-\pi$ and a ϕ value from a distribution of 10 evenly spaced values across the interval $0-2\pi$, with the endpoint of 2π not included.

Subsequent experiments set $\phi = 0$ as it was found that ϕ did not affect measurement outcomes; this decision is discussed further in Sections IV & V. These follow-on experiments also used 100 unique circuits. Each circuit then uniquely used a θ value from a distribution of 100 evenly spaced values across the interval $0 - \pi$.

B. Deriving Models for System Behaviour with Circuit Errors

For the following section, further details of how model derivations can be found in the Appendices in Section VIII.

Assuming SPAM and incoherent errors, we derived equations for the probability of measuring 0 for a given desired qubit state.

We started by defining a qubit with preparation errors and its corresponding density matrix:

$$|\psi\rangle = \cos(\frac{\theta+\epsilon}{2})|0\rangle + e^{i(\phi+\mu)}\sin(\frac{\theta+\epsilon}{2})|1\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$\rho_\psi = |\psi\rangle\langle\psi| = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix}$$

where ϵ and μ are preparation error parameters and $|\alpha|^2 + |\beta|^2 = 1$. We then defined a misaligned measurement apparatus and its corresponding density matrix:

$$|\tilde{0}\rangle = \cos\left(\frac{\nu}{2}\right)|0\rangle + e^{i\tau}\sin\left(\frac{\nu}{2}\right)|1\rangle = \gamma|0\rangle + \delta|1\rangle$$
$$\tilde{M}_0 := |\tilde{0}\rangle\langle\tilde{0}| = \begin{pmatrix} |\gamma|^2 & \gamma\delta^* \\ \gamma^*\delta & |\delta|^2 \end{pmatrix}$$

where ν and τ are measurement error parameters and $|\gamma|^2 + |\delta|^2 = 1$. Incoherent errors were then modelled using the following Kraus operators:

$$A_0 = \sqrt{p_I}I \quad A_1 = \sqrt{p_X}X \quad A_2 = \sqrt{p_Y}Y \quad A_3 = \sqrt{p_Z}Z$$

where p_σ is the probability the system evolves following σ, and the following must hold:

$$p_I + p_X + p_Y + p_Z = 1 \tag{1}$$

Using these equations, we derived Equation 2 to model the probability of measuring zero in the presence of circuit errors.

Assuming no SPAM phase errors and setting $\phi = 0$, we derived Equation 3 to model the system with state-independent circuit errors. These assumptions are discussed further in Sections IV & V.

We then derived models assuming state-dependent circuit errors by transforming the Kraus probabilities to depend on θ:

$$p_X \to x\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_Y \to y\sin^2\left(\frac{\theta+\epsilon}{2}\right)$$
$$p_Z \to z\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_I \to l\cos^2\left(\frac{\theta+\epsilon}{2}\right)$$

where $0 \leq \{x, y, z, l\} \leq 1$. Note that p_I did not use the parameter "i" to avoid confusion with complex numbers. Following this, we transformed the preparation error to also depend on θ:

$$\epsilon \to \epsilon\sin^2\left(\frac{\theta}{2}\right)$$

where $-\frac{\pi}{2} \leq \epsilon \leq \frac{\pi}{2}$. With these transformations, we derived Equations 4 & 5 to model the probability of measuring zero in the presence of state-dependent circuit errors.

When $\theta = 0$, Equation 4 simplifies to depend only on the measurement error parameter. That is $\nu =$

$2\cos^{-1}(\sqrt{\hat{\tilde{p}}\,(0\ |\theta=0)}) = \nu'$. Using this identity, we derived Equation 6. When $\theta = 0$, Equation 5 reduces to $l = 1$, using this statement, we derived Equation 7.

For completeness, we also model state-dependent circuit errors by alternatively transforming Kraus probability θ dependence:

$$p_X \to x\sin\left(\frac{\theta+\epsilon}{2}\right) \quad p_Y \to y\sin\left(\frac{\theta+\epsilon}{2}\right)$$
$$p_Z \to z\sin\left(\frac{\theta+\epsilon}{2}\right) \quad p_I \to l\cos\left(\frac{\theta+\epsilon}{2}\right)$$

where $0 \leq \{x, y, z, l\} \leq 1$. Note that p_I did not use a parameter "i" to avoid confusion with complex numbers. Following this, we alternatively transformed the preparation error θ dependence:

$$\epsilon \to \epsilon\sin\left(\frac{\theta}{2}\right)$$

With these alternative transformations, we derived Equations 8 & 9 to model the probability of measuring zero in the presence of state-dependent circuit errors.

With these alternative models, when $\theta = 0$, Equation 8 simplifies to depend only on the measurement error parameter. That is $\nu = 2\cos^{-1}(\sqrt{\hat{\tilde{p}}(0\ |\theta=0)}) = \nu'$. Using this identity, we derived Equation 10. When $\theta = 0$, Equation 9 reduces to $l = 1$, using this statement, we derived Equation 11.

$$\tilde{p}(0 \mid \theta \wedge \phi) = \frac{1}{2}\,[\,1 + \cos(\theta+\epsilon)\cos(\nu)(1-2p_X-2p_Y) + \sin(\theta+\epsilon)\sin(\nu)\,(\cos(\phi+\mu+\tau)(p_I-p_Z) + \cos(\phi+\mu-\tau)(p_X-p_Y))\,] \tag{2}$$

$$\tilde{p}(0 \mid \theta) = \frac{1}{2}\,(\,1 + (1-2p_X-2p_Y)\cos(\theta+\epsilon)\cos(\nu) + (1-2p_Z-2p_Y)\sin(\theta+\epsilon)\sin(\nu)\,) \tag{3}$$

$$\tilde{p}(0 \mid \theta) = \frac{1}{2}\,(\,1 + \cos(\nu)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,(\,1-x-y+(x+y)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,) + \sin(\nu)\sin\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,(\,1-y-z+(y+z)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,)\,) \tag{4}$$

$$l + x + y + z + (l-x-y-z)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right) = 2 \tag{5}$$

$$\tilde{p}(0 \mid \theta) = \frac{1}{2}\,[\,1 + \cos(\nu')\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,(\,1-x-y+(x+y)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,) + \sin(\nu')\sin\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,(\,1-y-z+(y+z)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\,)\,] \tag{6}$$

$$(x+y+z-1)\left(1-\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\right) = 0 \tag{7}$$

$$\tilde{p}(0 \mid \theta) = \frac{1}{2}\,[\,1 + \cos(\nu)\cos\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\,(\,1-2(x+y)\sin\left(\frac{1}{2}\left[\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right]\right)\,) + \sin(\nu)\sin\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\,(\,1-2(y+z)\sin\left(\frac{1}{2}\left[\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right]\right)\,)\,] \tag{8}$$

$$l\cos\left(\frac{1}{2}\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\right) + (x+y+z)\sin\left(\frac{1}{2}\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\right) = 1 \tag{9}$$

$$\tilde{p}(0 \mid \theta) = \frac{1}{2}\,[\,1 + \cos(\nu')\cos\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\,(\,1-2(x+y)\sin\left(\frac{1}{2}\left[\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right]\right)\,) + \sin(\nu')\sin\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\,(\,1-2(y+z)\sin\left(\frac{1}{2}\left[\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right]\right)\,)\,] \tag{10}$$

$$\cos\left(\frac{1}{2}\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\right) + (x+y+z)\sin\left(\frac{1}{2}\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\right) = 1 \tag{11}$$

TABLE I
PSO EXPERIMENT HYPERPARAMETERS

Cognitive Factor (c1)	Social Factor (c2)	Inertia (w)
0.5	0.3	0.9

C. PSO for Parameter Estimation

The PySwarms library [14] was used to implement the PSO algorithm. The PSO algorithm hyperparameters used for all experiments can be found in Table I, and all other hyperparameters set their default values. For all PSO experiments, the defined problem space was n-dimensional, where $n-1$ dimensions represented error parameter values and the n^{th} dimension represented fitness function values. The fitness function for all experiments was the MSE between two distributions of measurement statics.

1) Inferring Circuit Errors: To study circuit errors, we used six different models. The models were:

- state-independent errors using Equations 1 & 3
- state-dependent errors using Equations 4 & 5
- state-dependent errors using Equations 6 & 7
- state-dependent errors using Equations 8 & 9
- state-dependent errors using Equations 10 & 11

Specifically, we used the PSO algorithm to find circuit error parameters that resulted in the model being able to best replicate the experimental data and the inferred Kraus probabilities being as close to bounds. The fitness function for the PSO algorithm was the sum of the MSE between a given model's simulated probability data and experimental probability data and the distance between model Kraus probability and their corresponding bound. For these experiments, we used 50 particles and executed 5000 iterations. As is discussed further in Subsection IV-B, assuming state-dependent circuit errors using Equations 10 & 11 best modelled the experimental probability data. Thus, Equations 10 & 11 were used for subsequent experiments.

2) PSO Convergence Behaviour: We then studied PSO algorithm convergence by starting particles in different locations within the solution space. We used 25 particles and 2500 iterations for the convergence studies. For these experiments, all particles used the same initial value for one parameter, and the remaining parameters were randomised across the parameter's respective range. For example, when testing the effect of ϵ initialisation, all particles would share the same initial ϵ position, and x, y, z were randomised. Then for a given initial parameter value, after the 2500 iterations, the best solution for that particular initial value was recorded. A distribution of ten evenly spaced values across the interval of the parameter's maximum and minimum values was studied for each parameter. Five rounds were conducted for each initial value, and the reported solutions in Section IV were averaged across those five rounds. In total, 200 experiments were conducted as we tested four parameters with 10 initial values, and each experiment was repeated five times.

3) Control Parameters for Circuit Error Effect Mitigation: Our final study explored if circuit error effects could be mitigated using control parameters.

During these circuit error mitigation studies, we used the IBMQ Quito Simulator as experiments required frequently executing circuits. The execution queue for live quantum computers would have resulted in extensive experiment times; thus, it was unsuitable for these experiments.

For this study, we started by using the circuit error parameter inference techniques to find SPAM errors for the simulator. We used θ values from a distribution of 100 evenly spaced values across the interval of $0-\pi$. This paper reports the simulation circuit error parameters averaged across parameter initialisation convergence experiments.

We followed the inference of simulator circuit error parameters by updating the PSO techniques to find control parameters to mitigate the circuit error effects. We used the same 100 evenly spaced θ values over the range $0-\pi$ to find control parameters. However, before initialising the qubit with desired θ value, we would correct for preparation errors and update θ as follows:

$$\theta' = \theta + \epsilon \sin^2\left(\frac{\theta}{2}\right)$$

Before the qubit was measured, we would rotate the qubit to account for an imperfect measurement basis. We assumed a gate rotation error for these rotations and mitigated its effects using the preparation error control parameter. Thus, the corrective rotation angle for measurement error mitigation and subsequent matrix were derived using the following:

$$\nu' = \nu + \epsilon \sin^2\left(\frac{\nu}{2}\right)$$

$$\sigma_y(\nu') = \begin{pmatrix} \cos(\frac{\nu'}{2}) & -\sin(\frac{\nu'}{2}) \\ \sin(\frac{\nu'}{2}) & \cos(\frac{\nu'}{2}) \end{pmatrix}$$

Noise mitigation PSO experiments used five particles, and 500 iterations were executed. We significantly reduced the number of interactions and particles because of the high computation cost of simulation executions.

IV. RESULTS & DISCUSSION

This section will detail and briefly discuss each experiment's results. Note that throughout this section, when we refer to squared error and MSE, we are calculating the squared error between the theoretical ideal probability (denoted p) and the estimated probability (denoted $\hat{\hat{p}}$).

A. Measurement Statistics for Varyious θ and ϕ Values

The global phase of a qubit should not affect the measurement outcomes when measuring the qubit state relative to the computational basis. However, we still studied the qubit phase as we hypothesised that phase changes might introduce additional circuit errors that affect measurement outcomes. As Figures 3 & 4 illustrate, varying the phase value ϕ did not affect measurement outcomes. However, as Figure 3 demonstrates, there are no statistically significant differences

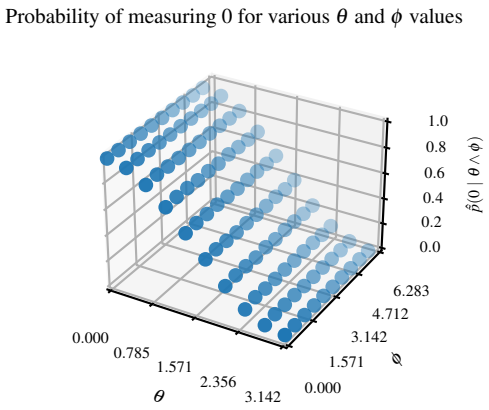

Fig. 3. The estimated probability of measuring zero ($\hat{\tilde{p}}$) assuming circuit errors for qubits with various θ and ϕ initialisation values. The probability is estimated by calculating the number of zeros measured (n_0) divided by the total number of executions (n), that is, $\hat{\tilde{p}}(0 \mid \theta \wedge \phi) := \frac{n_0}{n}$.

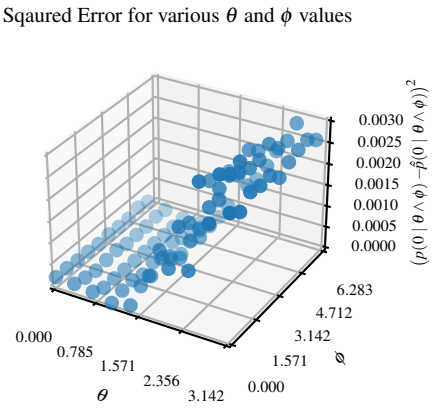

Fig. 4. The squared error between the theoretical probability of measuring zero (p) assuming no circuit errors and estimated probability of measuring zero ($\tilde{p}$) for qubits with various θ and ϕ initialisation values. The square error is calculated as $\left(p(0 \mid \theta \wedge \phi) - \hat{\tilde{p}}(0 \mid \theta \wedge \phi)\right)^2$.

in probability outcomes across phase values. We calculated a p-value of 1.0 using a one-way ANOVA [4] across the various phase value probability distributions. Corroborating Figure 3, Figure 4 illustrates there are no statistically significant differences in squared error across different phase values. We calculated a p-value of 0.9999 using a one-way ANOVA [4] across the various phase value square error distributions. As the qubit phase value did not affect measurement outcomes, we thus simplified our models by setting $\phi = 0$ and assumed no phase errors. Results from these simplified experiments are seen in Figures 5 & 6.

Though the qubit phase did not affect measurement outcomes, Figures 4 & 6 illustrate that as θ increases, the square error between ideal and measured probabilities also increases. This behaviour of error increasing as a function of θ motivated the state-dependent circuit error models.

Interestingly, Figures 4 & 6 show that squared errors were not minimal when $\theta = 0$ but were smallest when $\theta \approx \frac{\pi}{4}$. This interesting behaviour is discussed later in Section V.

B. Inferring Error Parameters

As Table II shows, assuming state-dependent errors with errors increasing following $\sin^2\left(\frac{\theta}{2}\right)$ could most accurately model the measured data. Specifically, the PSO algorithm could find circuit error parameters for Equations 6 & 7, which resulted in the smallest MSE between the model's simulated and the experimental probability data. This model could reduce the MSE between its simulated and the experimental probability data by 91.87% when compared to the ideal theoretical data. To further support Equations 6 & 7 as an appropriate model, the related Equations 4 & 5 (those which did no substitute for ν and l) were the second best at modelling experiment data.

All other models performed worse or marginally than the ideal model in replicating experiment data.

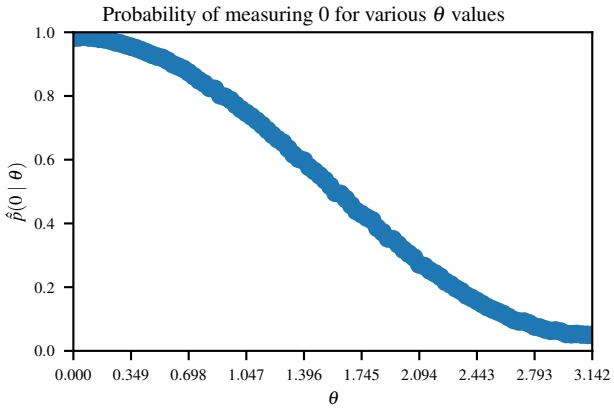

Fig. 5. The estimated probability of measuring zero ($\hat{\tilde{p}}$) assuming circuit errors for qubits with various θ initialisation values. The probability is estimated by calculating the number of zeros measured (n_0) divided by the total number of executions (n), that is, $\hat{\tilde{p}}(0 \mid \theta) := \frac{n_0}{n}$.

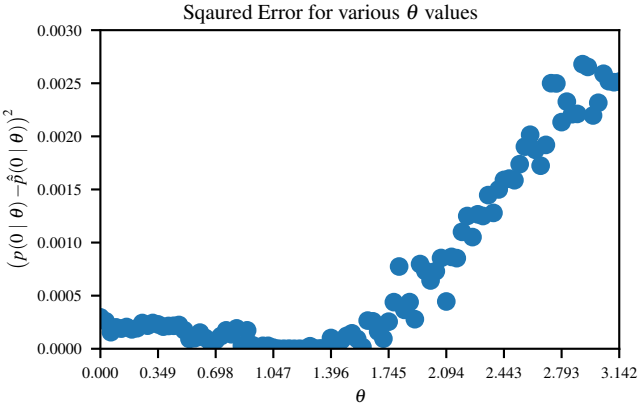

Fig. 6. The squared error between the theoretical probability of measuring zero (p) assuming no circuit errors and the estimated probability of measuring zero ($\tilde{p}$) for qubits with various θ initialisation values. The square error is calculated as $\left(p(0 \mid \theta) - \hat{\tilde{p}}(0 \mid \theta)\right)^2$.

TABLE II
DIFFERING MODEL DESCRIPTION AND PERFORMANCE

Model Description	Error model	Probability Equation	Kraus Probability Bounding Equation	Final MSE
Control (ideal data)	no error	$\cos^2\left(\frac{\theta}{2}\right)$	$p_I = 1$	0.000750
State-independent error	constant	3	1	0.001370
State-dependent error (not substituting for ν and l)	$\sin^2\left(\frac{\theta}{2}\right)$	4	5	0.000095
State-dependent error (substituting for ν and l)	$\mathbf{\sin^2\left(\frac{\theta}{2}\right)}$	**6**	**7**	**0.000061**
State-dependent error (not substituting for ν and l)	$\sin\left(\frac{\theta}{2}\right)$	8	9	0.000228
State-dependent error (substituting for ν and l)	$\sin\left(\frac{\theta}{2}\right)$	10	11	0.002567

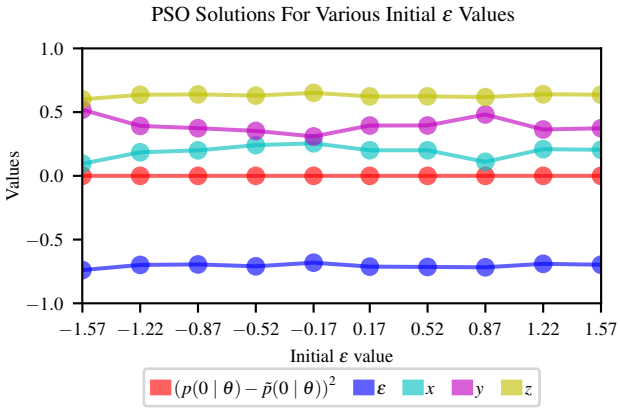

Fig. 7. PSO-inferred circuit error parameters over various initial ϵ values. All particles started with the same ϵ value, and all other parameter values were randomised across all the particles. The inferred circuit error parameters are displayed for each particular initial ϵ value.

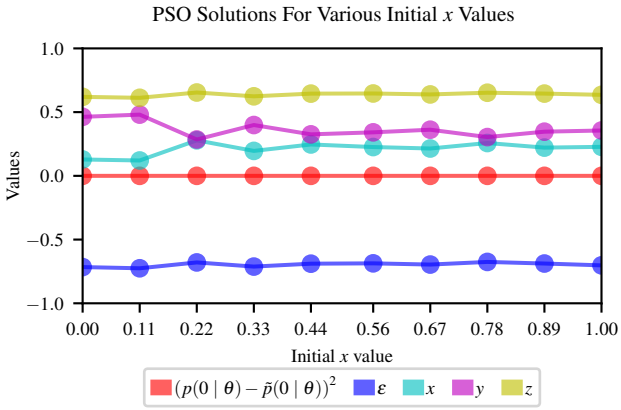

Fig. 8. PSO-inferred circuit error parameters over various initial x values. All particles started with the same x value, and all other parameter values were randomised across all the particles. The inferred circuit error parameters are displayed for each particular initial x value.

C. Different Initial Particle Positions

As Figures 7-10 demonstrate, regardless of initial particle positions, the PSO algorithm converged to approximately the same inferred circuit error parameters. The inferred solutions for ϵ and z are relatively constant. This consistent behaviour of ϵ and z solutions likely results from ν and l being set constant. This consistency suggests a relationship between ϵ, ν, z, l. We found a linear correlation between ϵ and z with a correlation coefficient of 0.9923, which further supports relationships between ϵ, ν, z, l. The inferred x and y values fluctuate more than the other error parameters. There is an anti-linear correlation between x and y with a correlation coefficient of -0.9834.

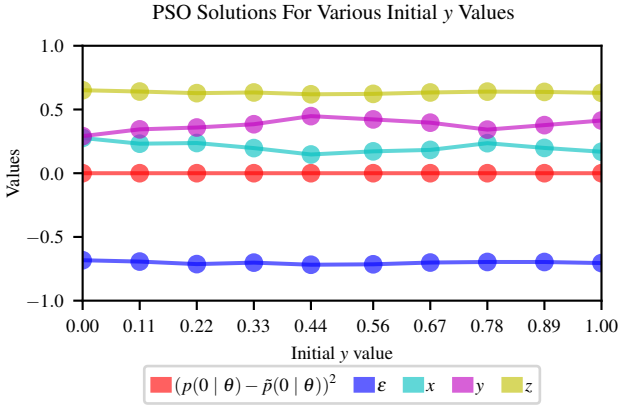

Fig. 9. PSO-inferred circuit error parameters over various initial y values. All particles started with the same y value, and all other parameter values were randomised across all the particles. The inferred circuit error parameters are displayed for each particular initial y value.

D. Control Parameters for Noise Minimisation

As Table IV demonstrates, PSO-inferred control parameters are effective as we reduce the effect of noise by 61.96%.

Table IV compares the inferred circuit error parameters and circuit error mitigation control parameters. As Table IV shows, using inferred circuit error parameters to mitigate errors is worse than not mitigating the circuit errors. We hypothesise these results occur as the PSO control parameters may also correct for incoherent errors, and therefore using them results in superior performance.

Finally, Table III demonstrates the similarity in the simulator and live quantum computer inferred circuit error parameters. This similarity suggests that the simulator implements realistic circuit errors. A realistic simulator is essential to assure researchers that algorithms developed and tested on the simulator would perform similarly on a live quantum computer.

TABLE III
AVERAGE INFERRED CIRCUIT ERROR PARAMETERS AND MSE

Source of Experiment Data	ϵ	ν	x	y	z	l	MSE
IMBQ Quito	-0.7062	0.2631	0.1944	0.3919	0.6299	1.0	0.000056
IMBQ Quito Simulator	-0.7418	0.3055	0.2253	0.3858	0.6065	1.0	0.000076

TABLE IV
MSE WITH VARIOUS CONTROL PARAMETERS

Method to Infer Control Parameter	Inferred ϵ	Inferred ν	MSE
No correction	0.0	0.0	0.001530
PSO-Inferred Control	**-0.2721**	**0.0749**	**0.000582**
PSO-Inferred Circuit Error	-0.7418	0.3055	0.001722

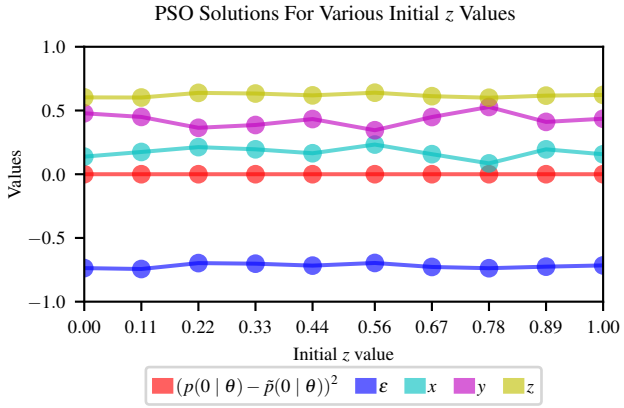

Fig. 10. PSO-inferred circuit error parameters over various initial z values. All particles started with the same z value, and all other parameter values were randomised across all the particles. The inferred circuit error parameters are displayed for each particular initial z value.

V. DISCUSSION

This paper's results show that the PSO algorithm can be used to infer circuit error parameters. Additionally, the PSO algorithm can be used to find control parameters that mitigate circuit error effects. More broadly, our experimental results support the application of ML techniques to address noise in quantum computers.

A. Assuming MSE Measures Model Representation Ability

We acknowledge that a fundamental assumption of our work is that the MSE between simulated and experimental data is a metric for the model's ability to represent the underlying physical processes. However, this assumption may be invalid as it may be the case that, for some models, it is easier for the PSO algorithm to optimise model parameters and achieve a lower MSE. Thus, a low MSE may indicate that the model may be easier to optimise rather than the model being an accurate representation of underlying physical processes.

However, to contradict this assumption, the best-performing model was arguably more complex as it contained more non-linear functions and parameter terms than the other models. Given the model's increased complexity, it is thus likely that its parameters are subsequently more difficult to optimise. Therefore, given the superior performance and yet increased complexity of Equations 6 & 7, this supports these equations as better models of the underlying physical process.

To further support state-dependent circuit errors, and thus Equations 6 & 7, it is physically sensible that preparation error and incoherent error probabilities increase as a function of θ and, thereby, the energy of the qubit. A larger θ would likely mean that preparation operations would be applied for longer, thus increasing the preparation error.

B. Assuming No Phase Errors

Another assumption we made is that qubits had no phase errors. In our current experiments, if phase errors were present, however, we would not have been able to detect them as the global phase of a qubit has no impact on measurements when projecting onto the computational basis.

However, one should consider phase errors as there would exist situations where the qubit phase should be precise, such as in the application of the Hadamard gate.

We initially hypothesised that like increasing θ, increasing ϕ would introduce more energy into the system and thus increase the chances of circuit error. Perhaps, it is the case that this happened, but the error was introduced into the phase of the qubit, which would not have been able to detect.

To study phase errors, ϕ could be set to 0, but models could assume phase errors. Thus, these circuit phase error parameters could be inferred alongside the other parameters.

C. Assuming Error Dependence on θ

We have also assumed that error depends on θ by a constant amount. It is reasonable to conjecture that error is better modelled using a normal distribution. When modelled by a normal distribution, increasing θ would increase the mean of this distribution.

Another possible reason error increases as a function of θ is that decoherence increases as a function of circuit executions. Higher θ values could be executed first to test this hypothesis, and experimentalists could study if the squared error relationship as a function of θ continues to hold. For all our experiments, higher θ values were executed later. Thus, it appears that the error is correlated with θ.

D. Position of Minimal Error

Interestingly, the minimal squared error occurs not when $\theta = 0$ but when $\theta \approx \frac{\pi}{4}$. We hypothesise that this value of θ results in some combination of circuit errors, which minimise and cancel each other out and thus, the squared error is minimal at this point.

E. Extending Experiment Results

Experiment results could be extended by:

- Repeating the PSO convergence experiments but not assuming values for ν and l. This extended experiment allows one to test for other relationships between parameters.
- Using alternative optimisation techniques to infer parameters. If similar solutions are still inferred using these settings, this supports global best parameter solutions if similar parameters are inferred with different optimisation techniques.
- Expanding the assumed range for SPAM errors. If similar solutions are still inferred using these settings, this supports global best parameter solutions if similar parameters are inferred with a wider parameter range.

VI. Conclusions

The application of ML to infer and mitigate circuit errors was studied in this paper. First, we studied circuit error over various θ and ϕ values by measuring the difference in predicted and measured probabilities. We found that differing ϕ values did not affect the squared error between ideal and measured distributions. However, increasing θ values did affect the squared error. Subsequently, we developed different equations to model measurement statistics assuming circuit errors. We then used the PSO algorithm to infer circuit error parameters for each model and measured how close to experimental data the model's generated data could match. We found that assuming state-dependent errors with error increase according to $sin^2\left(\frac{\theta}{2}\right)$ could model the experimental data the best. With this model, we started particles in different locations of the problem space and studied the algorithm convergence behaviour. We found that some inferred parameters remained almost constant, and others fluctuated within bounds. Finally, we measured the validity of using control parameters to mitigate circuit error effects. PSO-optimised control parameters could mitigate circuit error effects for a quantum computer simulation. Given the study's positive results, it is recommended that the application of ML techniques to infer and mitigate quantum computing noise continues.

VII. Future Work

One suggested direction for future work is studying phase errors to a greater degree. Phase errors could be studied by measuring the qubit along a different axis, such as the Hadamard axis and using techniques developed in this paper to infer circuit error parameters related to ϕ. However, one consideration for these experiments is that the Qiskit library allows only measurement along the computation basis. This Qiskit limitation can be mitigated by first rotating the qubit to sit along another measurement basis. However, these extra gate rotations may introduce other additional errors.

Another future project could acquire measurement statistics from different computers and timings. Then for each computer and timing, perform parameter inference and test if inferred pSPAM and incoherent error parameters are the same across machines and timings. It is hypothesised that a machine's SPAM error parameters are constant. However, incoherent error parameters are more randomised.

Another optimisation algorithm to use for experiments could be Bayesian optimisation. For Bayesian optimisation, there would be an n-dimensional space, where $n-1$ dimensions are error parameters, and the n^{th} dimension is the MSE fitness function.

Other performance metrics, such as total or max variation distance, could be used as optimisation fitness metrics. To enable a comparison for these experiments, all three metrics (MSE, total and max variation) should be measured at the end of the optimisation to see which fitness function resulted in the best performance across all three metrics.

Finally, the PSO-inferred control parameters techniques developed here could be easily transferred to a live quantum computer to prove the technique's utility further. The mitigations could be applied over a circuit with ever-increasing depth to test the robustness of the PSO-inferred control parameters.

References

[1] MD SAJID ANIS, Abby-Mitchell, Héctor Abraham ...., and Mantas Čepulkovskis. Qiskit: An open-source framework for quantum computing, 2021.

[2] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. Correcting coherent errors with surface codes. *npj Quantum Information*, 4(1):1–6, 2018.

[3] Lov K Grover. A framework for fast quantum mechanical algorithms. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 53–62, 1998.

[4] Gary W Heiman. *Understanding research methods and statistics: An integrated introduction for psychology*. Houghton, Mifflin and Company, 2001.

[5] Joseph K Iverson and John Preskill. Coherence in logical quantum channels. *New Journal of Physics*, 22(7):073066, 2020.

[6] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[7] Anastasios Kyrillidis. Introduction to quantum computing: bloch sphere. *URL: http://akyrillidis. github. io/notes/quant_post_7*, 2019.

[8] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy Lloyd O'Brien. Quantum computers. *nature*, 464(7285):45–53, 2010.

[9] Haggai Landa, Dekel Meirom, Naoki Kanazawa, Mattias Fitzpatrick, and Christopher J Wood. Experimental bayesian estimation of quantum state preparation, measurement, and gate errors in multiqubit devices. *Physical Review Research*, 4(1):013199, 2022.

[10] Kai Li. The qubit fidelity under different error mechanisms based on error correction threshold. *Frontiers in Physics*, page 315, 2022.

[11] Federico Marini and Beata Walczak. Particle swarm optimization (pso). a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149:153–165, 2015.

[12] David McMahon. *Quantum computing explained*. John Wiley & Sons, 2007.

[13] N David Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007.

[14] Lester James V. Miranda. PySwarms, a research-toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, 3, 2018.
[15] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
[16] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
[17] Maximilian A Schlosshauer. *Decoherence: and the quantum-to-classical transition*. Springer Science & Business Media, 2007.
[18] Yuhui Shi and Russell C Eberhart. Parameter selection in particle swarm optimization. In *International conference on evolutionary programming*, pages 591–600. Springer, 1998.
[19] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

VIII. Appendices

A. Appendix 1: deriving probability equation with SPAM and incoherent errors

We start by defining a qubit with preparation errors and its corresponding density matrix:

$$|\psi\rangle = \cos(\frac{\tilde{\theta}_\psi}{2})|0\rangle + e^{i\tilde{\phi}_\psi}\sin(\frac{\tilde{\theta}_\psi}{2})|1\rangle = \cos(\frac{\theta+\epsilon}{2})|0\rangle + e^{i(\phi+\mu)}\sin(\frac{\theta+\epsilon}{2})|1\rangle = \alpha|0\rangle + \beta|1\rangle$$
$$\tilde{\rho}_\psi = |\psi\rangle\langle\psi| = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix}$$

where ϵ and μ are preparation error parameters and $|\alpha|^2 + |\beta|^2 = 1$. We then define a misaligned measurement apparatus and its corresponding density matrix:

$$|\tilde{0}\rangle = \cos\left(\frac{\theta_{\tilde{0}}}{2}\right)|0\rangle + e^{i\phi_{\tilde{0}}}\sin\left(\frac{\theta_{\tilde{0}}}{2}\right)|1\rangle = \cos\left(\frac{\nu}{2}\right)|0\rangle + e^{i\tau}\sin\left(\frac{\nu}{2}\right)|1\rangle = \gamma|0\rangle + \delta|1\rangle$$
$$\tilde{M}_0 := |\tilde{0}\rangle\langle\tilde{0}| = \begin{pmatrix} |\gamma|^2 & \gamma\delta^* \\ \gamma^*\delta & |\delta|^2 \end{pmatrix}$$

where ν and τ are measurement error parameters and $|\gamma|^2 + |\delta|^2 = 1$. Incoherent errors were then modelled using the following Kraus operators:

$$A_0 = \sqrt{p_I}I \quad A_1 = \sqrt{p_X}X \quad A_2 = \sqrt{p_Y}Y \quad A_3 = \sqrt{p_Z}Z$$

where p_σ is the probability the systems evolve under the given σ, and

$$\begin{aligned}\sum_{k=0}^{4} A_k A_k^\dagger &= p_I I + p_X X X^\dagger + p_Y Y Y^\dagger + p_Z Z Z^\dagger = I \\ &= p_I I + p_X I + p_Y I + p_Z I = I \\ &= I(p_I + p_X + p_Y + p_Z) = I \\ &= p_I + p_X + p_Y + p_Z = 1\end{aligned}$$

The depolarisation channel is defined as:

$$\begin{aligned}\phi(\rho) &= \sum_{k=0}^{4} A_k \rho A_k^\dagger \\ &= p_I I\rho I + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z \\ &= p_I\rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z\end{aligned}$$

A qubit's evolution through this depolarisation channel is then:

$$\tilde{\rho}_\psi = \phi(\rho_\psi) = p_I\rho_\psi + p_X X\rho_\psi X + p_Y Y\rho_\psi Y + p_Z Z\rho_\psi Z$$

$$= p_I \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix} + p_X \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$+ p_Y \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} + p_Z \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$= p_I \begin{pmatrix} |\alpha|^2 & \alpha\beta^* \\ \alpha^*\beta & |\beta|^2 \end{pmatrix} + p_X \begin{pmatrix} |\beta|^2 & \alpha^*\beta \\ \alpha\beta^* & |\alpha|^2 \end{pmatrix} + p_Y \begin{pmatrix} |\beta|^2 & -\alpha^*\beta \\ -\alpha\beta^* & |\alpha|^2 \end{pmatrix} + p_Z \begin{pmatrix} |\alpha|^2 & -\alpha\beta^* \\ -\alpha^*\beta & |\beta|^2 \end{pmatrix}$$

$$= \begin{pmatrix} p_I|\alpha|^2 & p_I\alpha\beta^* \\ p_I\alpha^*\beta & p_I|\beta|^2 \end{pmatrix} + \begin{pmatrix} p_X|\beta|^2 & p_X\alpha^*\beta \\ p_X\alpha\beta^* & p_X|\alpha|^2 \end{pmatrix} + \begin{pmatrix} p_Y|\beta|^2 & -p_Y\alpha^*\beta \\ -p_Y\alpha\beta^* & p_Y|\alpha|^2 \end{pmatrix} + \begin{pmatrix} p_Z|\alpha|^2 & -p_Z\alpha\beta^* \\ -p_Z\alpha^*\beta & p_Z|\beta|^2 \end{pmatrix}$$

$$= \begin{pmatrix} p_I|\alpha|^2 + p_X|\beta|^2 + p_Y|\beta|^2 + p_Z|\alpha|^2 & p_I\alpha\beta^* + p_X\alpha^*\beta - p_Y\alpha^*\beta - p_Z\alpha\beta^* \\ p_I\alpha^*\beta + p_X\alpha\beta^* - p_Y\alpha\beta^* - p_Z\alpha^*\beta & p_I|\beta|^2 + p_X|\alpha|^2 + p_Y|\alpha|^2 + p_Z|\beta|^2 \end{pmatrix}$$

$$= \begin{pmatrix} (p_I + p_Z)|\alpha|^2 + (p_X + p_Y)|\beta|^2 & (p_I - p_Z)\alpha\beta^* + (p_X - p_Y)\alpha^*\beta \\ (p_I - p_Z)\alpha^*\beta + (p_X - p_Y)\alpha\beta^* & (p_X + p_Y)|\alpha|^2 + (p_I + p_Z)|\beta|^2 \end{pmatrix}$$

Probability of measuring zero is $\tilde{p}(0) = Tr(\tilde{\rho}_\psi \tilde{M}_0) = Tr(\tilde{\rho}_\psi|\tilde{0}\rangle\langle\tilde{0}|)$. Let us first consider $\tilde{\rho}_\psi|\tilde{0}\rangle\langle\tilde{0}|$:

$$= \begin{pmatrix} (p_I + p_Z)|\alpha|^2 + (p_X + p_Y)|\beta|^2 & (p_I - p_Z)\alpha\beta^* + (p_X - p_Y)\alpha^*\beta \\ (p_I - p_Z)\alpha^*\beta + (p_X - p_Y)\alpha\beta^* & (p_X + p_Y)|\alpha|^2 + (p_I + p_Z)|\beta|^2 \end{pmatrix} \begin{pmatrix} |\gamma|^2 & \gamma\delta^* \\ \gamma^*\delta & |\delta|^2 \end{pmatrix}$$

$$= \begin{pmatrix} |\gamma|^2 \left((p_I + p_Z)|\alpha|^2 + (p_X + p_Y)|\beta|^2 \right) & \gamma\delta^* \left[(p_I + p_Z)|\alpha|^2 + (p_X + p_Y)|\beta|^2 \right] \\ +\gamma^*\delta \left[(p_I - p_Z)\alpha\beta^* + (p_X - p_Y)\alpha^*\beta \right] & +|\delta|^2 \left[(p_I - p_Z)\alpha\beta^* + (p_X - p_Y)\alpha^*\beta \right] \\ \\ |\gamma|^2 \left[(p_I - p_Z)\alpha^*\beta + (p_X - p_Y)\alpha\beta^* \right] & \gamma\delta^* \left[(p_I - p_Z)\alpha^*\beta + (p_X - p_Y)\alpha\beta^* \right] \\ +\gamma^*\delta \left[(p_X + p_Y)|\alpha|^2 + (p_I + p_Z)|\beta|^2 \right] & +|\delta|^2 \left[(p_X + p_Y)|\alpha|^2 + (p_I + p_Z)|\beta|^2 \right] \end{pmatrix}$$

$\therefore p(0) = Tr(\tilde{\rho}_\psi|\tilde{0}\rangle\langle\tilde{0}|) =$

$$\begin{aligned} &|\gamma|^2[(p_I + p_Z)|\alpha|^2 + (p_X + p_Y)|\beta|^2] + \gamma^*\delta\left[(p_I - p_Z)\alpha\beta^* + (p_X - p_Y)\alpha^*\beta\right] \\ &+ \gamma\delta^*\left[(p_I - p_Z)\alpha^*\beta + (p_X - p_Y)\alpha\beta^*\right] + |\delta|^2\left[(p_X + p_Y)|\alpha|^2 + (p_I + p_Z)|\beta|^2\right] \end{aligned}$$

Let us consider first:

$$\begin{aligned} &|\gamma|^2\left[(p_I + p_Z)|\alpha|^2 + (p_X + p_Y)|\beta|^2\right] + |\delta|^2\left[(p_X + p_Y)|\alpha|^2 + (p_I + p_Z)|\beta|^2\right] \\ &= (p_I + p_Z)|\alpha|^2|\gamma|^2 + (p_X + p_Y)|\beta|^2|\gamma|^2 + (p_X + p_Y)|\alpha|^2|\delta|^2 + (p_I + p_Z)|\beta|^2|\delta|^2 \\ &= (p_I + p_Z)(|\alpha|^2|\gamma|^2 + |\beta|^2|\delta|^2) + (p_X + p_Y)(|\alpha|^2|\delta|^2 + |\beta|^2|\gamma|^2) \end{aligned}$$

Recall that:

$$|\alpha|^2 = \frac{1 + \cos(\tilde{\theta}_\psi)}{2} \quad |\beta|^2 = \frac{1 - \cos(\tilde{\theta}_\psi)}{2} \quad |\gamma|^2 = \frac{1 + \cos(\theta_{\tilde{0}})}{2} \quad |\delta|^2 = \frac{1 - \cos(\theta_{\tilde{0}})}{2}$$

Thus the equation becomes:

$$
\begin{aligned}
&(p_I + p_Z)\left(\left(\frac{1+\cos(\tilde{\theta}_\psi)}{2}\right)\left(\frac{1+\cos(\theta_{\tilde{0}})}{2}\right)+\left(\frac{1-\cos(\tilde{\theta}_\psi)}{2}\right)\left(\frac{1-\cos(\theta_{\tilde{0}})}{2}\right)\right)\\
&+(p_X + p_Y)\left(\left(\frac{1+\cos(\tilde{\theta}_\psi)}{2}\right)\left(\frac{1-\cos(\theta_{\tilde{0}})}{2}\right)+\left(\frac{1-\cos(\tilde{\theta}_\psi)}{2}\right)\left(\frac{1+\cos(\theta_{\tilde{0}})}{2}\right)\right)\\
&= (p_I + p_Z)\left(\cos^2(\frac{\tilde{\theta}_\psi}{2})\cos^2(\frac{\theta_{\tilde{0}}}{2})+\sin^2(\frac{\tilde{\theta}_\psi}{2})\sin^2(\frac{\theta_{\tilde{0}}}{2})\right)+(p_X + p_Y)\left(\cos^2(\frac{\tilde{\theta}_\psi}{2})\sin^2(\frac{\theta_{\tilde{0}}}{2})+\sin^2(\frac{\tilde{\theta}_\psi}{2})\cos^2(\frac{\theta_{\tilde{0}}}{2})\right)\\
&= \frac{p_I + p_Z}{2}\left(1+\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})\right)+\frac{p_X + p_Y}{2}\left(1-\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})\right)\\
&= \frac{1}{2}\left[\,(p_I + p_Z)(1+\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}}))+(p_X + p_Y)(1-\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}}))\,\right]\\
&= \frac{1}{2}\left[\,p_I + p_Z + p_X + p_Y + p_I\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})+p_Z\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})-p_Y\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})-p_X\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})\,\right]\\
&= \frac{1}{2}\left[\,p_I + p_Z + p_X + p_Y + \cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})(p_I + p_Z - p_Y - p_X)\,\right]
\end{aligned}
$$

Since $p_I + p_Z + p_X + p_Y = 1 \Rightarrow p_I + p_Z = 1 - p_X - p_Y$, this simplies the equation to:

$$\frac{1}{2}\left[1+\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})(1-2p_Y-2p_X)\right]$$

Now consider,

$$
\begin{aligned}
&\gamma^*\delta\left[(p_I - p_Z)\alpha\beta^* + (p_X - p_Y)\alpha^*\beta\right]+\gamma\delta^*\left[(p_I - p_Z)\alpha^*\beta + (p_X - p_Y)\alpha\beta^*\right]\\
&= (p_I - p_Z)\alpha\beta^*\gamma^*\delta + (p_X - p_Y)\alpha^*\beta\gamma^*\delta + (p_I - p_Z)\alpha^*\beta\gamma\delta^* + (p_X - p_Y)\alpha\beta^*\gamma\delta^*\\
&= (p_I - p_Z)(\alpha\beta^*\gamma^*\delta + \alpha^*\beta\gamma\delta^*) + (p_X - p_Y)(\alpha^*\beta\gamma^*\delta + \alpha\beta^*\gamma\delta^*)
\end{aligned}
$$

Given Bloch sphere notation of qubit and measurement operator $\alpha, \gamma \in \mathbb{R} \therefore \alpha = \alpha^*$ and $\gamma = \gamma^*$; simplify the equation to:

$$
\begin{aligned}
&(p_I - p_Z)(\alpha\beta^*\gamma\delta + \alpha\beta\gamma\delta^*) + (p_X - p_Y)(\alpha\beta\gamma\delta + \alpha\beta^*\gamma\delta^*)\\
&= \alpha\gamma[(p_I - p_Z)(\beta^*\delta + \beta\delta^*) + (p_X - p_Y)(\beta\delta + \beta^*\delta^*)]
\end{aligned}
$$

Recall that:

$$\alpha = \cos(\frac{\tilde{\theta}_\psi}{2}) \quad \beta = e^{i\tilde{\phi}_\psi}\sin(\frac{\tilde{\theta}_\psi}{2}) \quad \gamma = \cos(\frac{\theta_{\tilde{0}}}{2}) \quad \delta = e^{i\phi_{\tilde{0}}}\sin(\frac{\theta_{\tilde{0}}}{2})$$

Using these identities, the probability equation becomes:

$$\cos(\frac{\tilde{\theta}_\psi}{2})\cos(\frac{\theta_{\tilde{0}}}{2})\,[\,(p_I-p_Z)(e^{-i\tilde{\phi}_\psi}\sin(\frac{\tilde{\theta}_\psi}{2})e^{\phi_{\tilde{0}}}\sin(\frac{\theta_{\tilde{0}}}{2})+e^{i\tilde{\phi}_\psi}\sin(\frac{\tilde{\theta}_\psi}{2})e^{-i\phi_{\tilde{0}}}\sin(\frac{\theta_{\tilde{0}}}{2}))$$
$$+(p_X-p_Y)(e^{i\tilde{\phi}_\psi}\sin(\frac{\tilde{\theta}_\psi}{2})e^{i\phi_{\tilde{0}}}\sin(\frac{\theta_{\tilde{0}}}{2})+e^{-i\tilde{\phi}_\psi}\sin(\frac{\tilde{\theta}_\psi}{2})e^{-i\phi_{\tilde{0}}}\sin(\frac{\theta_{\tilde{0}}}{2}))\,]$$

$$=\cos(\frac{\tilde{\theta}_\psi}{2})\cos(\frac{\theta_{\tilde{0}}}{2})\,[\,\sin(\frac{\tilde{\theta}_\psi}{2})\sin(\frac{\theta_{\tilde{0}}}{2})(p_I-p_Z)(e^{-i\tilde{\phi}_\psi}e^{i\phi_{\tilde{0}}}+e^{i\tilde{\phi}_\psi}e^{-i\phi_{\tilde{0}}})+\sin(\frac{\tilde{\theta}_\psi}{2})\sin(\frac{\theta_{\tilde{0}}}{2})(p_X-p_Y)(e^{i\tilde{\phi}_\psi}e^{i\phi_{\tilde{0}}}+e^{-i\tilde{\phi}_\psi}e^{-i\phi_{\tilde{0}}})\,]$$

$$=\cos(\frac{\tilde{\theta}_\psi}{2})\cos(\frac{\theta_{\tilde{0}}}{2})\sin(\frac{\tilde{\theta}_\psi}{2})\sin(\frac{\theta_{\tilde{0}}}{2})\,[\,(p_I-p_Z)(e^{-i(\tilde{\phi}_\psi-\phi_{\tilde{0}})}+e^{i(\tilde{\phi}_\psi-\phi_{\tilde{0}})})+(p_X-p_Y)(e^{i(\tilde{\phi}_\psi+\phi_{\tilde{0}})}+e^{-i(\tilde{\phi}_\psi+\phi_{\tilde{0}})})\,]$$

$$=\cos(\frac{\tilde{\theta}_\psi}{2})\cos(\frac{\theta_{\tilde{0}}}{2})\sin(\frac{\tilde{\theta}_\psi}{2})\sin(\frac{\theta_{\tilde{0}}}{2})\,[\,2\cos(\tilde{\phi}_\psi-\phi_{\tilde{0}})(p_I-p_Z)+2\cos(\tilde{\phi}_\psi+\phi_{\tilde{0}})(p_X-p_Y)\,]$$

$$=\frac{2\sin(\tilde{\theta}_\psi)\sin(\theta_{\tilde{0}})}{4}\,[\,\cos(\tilde{\phi}_\psi+\phi_{\tilde{0}})(p_I-p_Z)+\cos(\tilde{\phi}_\psi-\phi_{\tilde{0}})(p_X-p_Y)\,]$$

$$=\frac{\sin(\tilde{\theta}_\psi)\sin(\theta_{\tilde{0}})}{2}\,[\,\cos(\tilde{\phi}_\psi+\phi_{\tilde{0}})(p_I-p_Z)+\cos(\tilde{\phi}_\psi-\phi_{\tilde{0}})(p_X-p_Y)\,]$$

Putting the two equations together, $\tilde{p}(0)$ =

$$\frac{1}{2}\,[\,1+\cos(\tilde{\theta}_\psi)\cos(\theta_{\tilde{0}})(1-2p_Y-2p_X)+\sin(\tilde{\theta}_\psi)\sin(\theta_{\tilde{0}})\,(\,\cos(\tilde{\phi}_\psi+\phi_{\tilde{0}})(p_I-p_Z)+\cos(\tilde{\phi}_\psi-\phi_{\tilde{0}})(p_X-p_Y)\,)\,]$$

Finally,

$$\tilde{\theta}_\psi=\theta+\epsilon\quad\tilde{\phi}_\psi=\phi+\mu\quad\theta_{\tilde{0}}=\nu\quad\phi_{\tilde{0}}=\tau$$

Substituting these values:

$$\frac{1}{2}\,[\,1+\cos(\theta+\epsilon)\cos(\nu)(1-2p_X-2p_Y)+\sin(\theta+\epsilon)\sin(\nu)\,(\,\cos(\phi+\mu+\tau)(p_I-p_Z)+\cos(\phi+\mu-\tau)(p_X-p_Y)\,)\,]$$

B. Appendix 2: deriving probability equation with state-independent SPAM and incoherent errors assuming no phases error

Substituting ϕ, μ, and τ = 0 into the probability equation from Subsection VIII-A:

$$\frac{1}{2}\,[\,1+\cos(\theta+\epsilon)\cos(\nu)(1-2p_X-2p_Y)+\sin(\theta+\epsilon)\sin(\nu)\,(\cos(0)(p_I-p_Z)+\cos(0)(p_X-p_Y))\,]$$
$$=\frac{1}{2}\,[\,1+\cos(\theta+\epsilon)\cos(\nu)(1-2p_X-2p_Y)+\sin(\theta+\epsilon)\sin(\nu)\,(p_I+p_X-p_Z-p_Y)\,]$$

Since $p_I+p_Z+p_X+p_Y=1\Rightarrow p_I+p_X=1-p_Z-p_Y$, this simplifies the equation to:

$$[\,1+(1-2p_X-2p_Y)\cos(\theta+\epsilon)\cos(\nu)+(1-2p_Z-2p_Y)\sin(\theta+\epsilon)\sin(\nu)\,]$$

C. Appendix 3: deriving probability equation with state-dependent SPAM and incoherent errors assuming no phases error

First assume Kraus probability dependence on $\tilde{\theta}_\psi = \theta + \epsilon$. This dependence is modelled as follows:

$$p_X = x\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_Y = y\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_Z = z\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_I = l\cos^2\left(\frac{\theta+\epsilon}{2}\right)^*$$

*Note that p_I did not use the parameter i to avoid confusion with complex numbers.

Substituting these relations into the probability equation from Subsection VIII-B:

$$\begin{aligned}
&\frac{1}{2}\left[\,1+\cos(\theta+\epsilon)\cos(\nu)\left(\,1-2x\sin^2\left(\frac{\theta+\epsilon}{2}\right)-2y\sin^2\left(\frac{\theta+\epsilon}{2}\right)\right)+\sin(\theta+\epsilon)\sin(\nu)\left(\,1-2z\sin^2\left(\frac{\theta+\epsilon}{2}\right)-2y\sin^2\left(\frac{\theta+\epsilon}{2}\right)\right)\right]\\
&=\frac{1}{2}\left[\,1+\cos(\theta+\epsilon)\cos(\nu)\left(\,1-2\sin^2\left(\frac{\theta+\epsilon}{2}\right)(x+y)\right)+\sin(\theta+\epsilon)\sin(\nu)\left(\,1-2\sin^2\left(\frac{\theta+\epsilon}{2}\right)(y+z)\right)\right]\\
&=\frac{1}{2}\left[\,1+\cos(\theta+\epsilon)\cos(\nu)\left(\,1-\frac{2(1-\cos(\theta+\epsilon))}{2}(x+y)\right)+\sin(\theta+\epsilon)\sin(\nu)\left(\,1-\frac{2(1-\cos(\theta+\epsilon))}{2}(y+z)\right)\right]\\
&=\frac{1}{2}\left[\,1+\cos(\theta+\epsilon)\cos(\nu)\left(\,1-(1-\cos(\theta+\epsilon))(x+y)\right)+\sin(\theta+\epsilon)\sin(\nu)\left(\,1-(1-\cos(\theta+\epsilon))(y+z)\right)\right]\\
&=\frac{1}{2}\left[\,1+\cos(\theta+\epsilon)\cos(\nu)\left(\,1-x-y+(x+y)\cos(\theta+\epsilon)\right)+\sin(\theta+\epsilon)\sin(\nu)\left(\,1-y-z+(y+z)\cos(\theta+\epsilon)\right)\right]
\end{aligned}$$

Finally, assume preparation error is dependent on θ. This dependence is modelled as follows:

$$\epsilon \to \epsilon\sin^2\left(\frac{\theta}{2}\right)$$

Substituting this relationship into the probability equation:

$$\begin{aligned}
&\frac{1}{2}\left(\,1+\cos(\nu)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\left(\,1-x-y+(x+y)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\right)\right.\\
&\qquad\left.+\sin(\nu)\sin\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\left(\,1-y-z+(y+z)\cos\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\right)\right)
\end{aligned}$$

D. Appendix 4: simplification of state-dependent Kraus probabilities assuming no phase errors

From Subsection VIII-C, the Kraus probabilities are modelled as:

$$p_X = x\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_Y = y\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_Z = z\sin^2\left(\frac{\theta+\epsilon}{2}\right) \quad p_I = l\cos^2\left(\frac{\theta+\epsilon}{2}\right)$$

From Subsection VIII-A, the following must hold:

$$p_X + p_Y + p_Z + p_I = 1$$

Substituting these state-dependent Kraus probabilities into the above bounding condition:

$$\begin{aligned}
&x\sin^2\left(\frac{\theta+\epsilon}{2}\right)+y\sin^2\left(\frac{\theta+\epsilon}{2}\right)+z\sin^2\left(\frac{\theta+\epsilon}{2}\right)+l\cos^2\left(\frac{\theta+\epsilon}{2}\right)=1\\
&=(x+y+z)\sin^2\left(\frac{\theta+\epsilon}{2}\right)+l\cos^2\left(\frac{\theta+\epsilon}{2}\right)=1\\
&=(x+y+z)\left(\frac{1-\cos(\theta+\epsilon)}{2}\right)+l\left(\frac{1+\cos(\theta+\epsilon)}{2}\right)=1\\
&=(x+y+z)(1-\cos(\theta+\epsilon))+l\,(1+\cos(\theta+\epsilon))=2\\
&=l+x+y+z+(l-x-y-z)\cos(\theta+\epsilon)=2
\end{aligned}$$

Finally, preparation error is transformed:

$$\epsilon \to \epsilon\sin^2\left(\frac{\theta}{2}\right)$$

Transforming the probability bounding equation to:

$$l + x + y + z + (l - x - y - z)\cos(\theta + \epsilon \sin^2\left(\frac{\theta}{2}\right)) = 2$$

E. Appendix 5: finding ν & l when $\theta = 0$ using probability equation with state-dependent SPAM and incoherent errors assuming no phase error

Substituting $\theta = 0$ into the Kraus probability bounding equation from Subsection VIII-D:

$$\begin{aligned}
2 &= l + x + y + z + (l - x - y - z)\cos(0 + \epsilon \sin^2\left(\frac{0}{2}\right)) \\
2 &= l + x + y + z + (l - x - y - z) \\
2 &= l + x + y + z + l - x - y - z \\
2 &= 2l \\
l &= 1
\end{aligned}$$

Substituting $l = 1$ into the Kraus probability bounding equation from Subsection VIII-D:

$$\begin{aligned}
2 &= 1 + x + y + z + (1 - x - y - z)\cos(\theta + \epsilon \sin^2\left(\frac{\theta}{2}\right)) \\
0 &= -1 + x + y + z + (1 - x - y - z)\cos(\theta + \epsilon \sin^2\left(\frac{\theta}{2}\right)) \\
0 &= (x + y + z - 1) - (x + y + z - 1)\cos(\theta + \epsilon \sin^2\left(\frac{\theta}{2}\right)) \\
0 &= (x + y + z - 1)(1 - \cos(\theta + \epsilon \sin^2\left(\frac{\theta}{2}\right))
\end{aligned}$$

Substituting $\theta = 0$ into the probability equation from Subsection VIII-C:

$$\begin{aligned}
&\frac{1}{2}\,[\,1 + \cos(\nu)\cos(0 + \epsilon \sin^2\left(\frac{0}{2}\right))\,(\,1 - x - y + (x + y)\cos(0 + \epsilon \sin^2\left(\frac{0}{2}\right))\,) \\
&\qquad + \sin(\nu)\sin(0 + \epsilon \sin^2\left(\frac{0}{2}\right))\,(\,1 - y - z + (y + z)\cos(0 + \epsilon \sin^2\left(\frac{0}{2}\right))\,)\,] \\
&= \frac{1}{2}\,[\,1 + \cos(\nu)\,(\,1 - x - y + (x + y)\,) + \sin(\nu)(0)\,(\,1 - y - z + (y + z)\,)\,] \\
&= \frac{1}{2}\,(\,1 + \cos(\nu)\,) \\
&= \cos^2(\frac{\nu}{2}) \\
\therefore \nu &= 2\cos^{-1}(\sqrt{\tilde{p}(0\ |\theta = 0)})
\end{aligned}$$

F. Appendix 6: deriving alternative probability equation with state-dependent SPAM and incoherent errors assuming no phase errors

First assume Kraus probability dependence on $\tilde{\theta}_\psi = \theta + \epsilon$. This dependence is modelled as follows:

$$p_X = x \sin\left(\frac{\theta + \epsilon}{2}\right) \quad p_Y = y \sin\left(\frac{\theta + \epsilon}{2}\right) \quad p_Z = z \sin\left(\frac{\theta + \epsilon}{2}\right) \quad p_I = l \cos\left(\frac{\theta + \epsilon}{2}\right)^*$$

*Note that p_I did not use the parameter i to avoid confusion with complex numbers. Substituting these relations into the probability equation from Subsection VIII-C:

$$\frac{1}{2}\,[\,1+\cos(\theta+\epsilon)\cos(\nu)\,(\,1-2x\sin\left(\frac{\theta+\epsilon}{2}\right)-2y\sin\left(\frac{\theta+\epsilon}{2}\right)\,)$$
$$+\sin(\theta+\epsilon)\sin(\nu)\,(\,1-2z\sin\left(\frac{\theta+\epsilon}{2}\right)-2y\sin\left(\frac{\theta+\epsilon}{2}\right))\,]$$

$$=\frac{1}{2}\,[\,1+\cos(\theta+\epsilon)\cos(\nu)\,(\,1-2(x+y)\sin\left(\frac{\theta+\epsilon}{2}\right)\,)$$
$$+\sin(\theta+\epsilon)\sin(\nu)\,(\,1-2(y+z)\sin\left(\frac{\theta+\epsilon}{2}\right)\,))\,]$$

Finally, assume preparation error is dependent on θ. This dependence is modelled as follows:

$$\epsilon\rightarrow\epsilon\sin\left(\frac{\theta}{2}\right)$$

Substituting this relationship into the probability equation:

$$\frac{1}{2}\,[1+\cos(\nu)\cos\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\,(\,1-2(x+y)\sin\left(\frac{1}{2}\left[\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right]\right)\,)$$
$$+\sin(\nu)\sin\left(\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right)\,(\,1-2(y+z)\sin\left(\frac{1}{2}\left[\theta+\epsilon\sin\left(\frac{\theta}{2}\right)\right]\right)\,)\,]$$

G. Appendix 7: alternative simplification of state-dependent Kraus probabilities assuming no phase errors

From Subsection VIII-F, the Kraus probabilities are modelled as:

$$p_X=x\sin\left(\frac{\theta+\epsilon}{2}\right)\quad p_Y=y\sin\left(\frac{\theta+\epsilon}{2}\right)\quad p_Z=z\sin\left(\frac{\theta+\epsilon}{2}\right)\quad p_I=l\cos\left(\frac{\theta+\epsilon}{2}\right)$$

From Subsection VIII-A, the following must hold:

$$p_X+p_Y+p_Z+p_I=1$$

Substituting these state-dependent Kraus probabilities into the above bounding condition:

$$x\sin\left(\frac{\theta+\epsilon}{2}\right)+y\sin\left(\frac{\theta+\epsilon}{2}\right)+z\sin\left(\frac{\theta+\epsilon}{2}\right)+l\cos\left(\frac{\theta+\epsilon}{2}\right)=1$$
$$=(x+y+z)\sin\left(\frac{\theta+\epsilon}{2}\right)+l\cos\left(\frac{\theta+\epsilon}{2}\right)=1$$

Finally, preparation error is transformed:

$$\epsilon\rightarrow\epsilon\sin^2\left(\frac{\theta}{2}\right)$$

Transforming the probability bounding equation to:

$$=(x+y+z)\sin\left(\frac{1}{2}\left(\theta+\epsilon\sin^2\left(\frac{\theta}{2}\right)\right)\right)+l\cos\left(\frac{1}{2}\left(\theta+\epsilon\cos^2\left(\frac{\theta}{2}\right)\right)\right)=1$$