QEC for GAD on QuDits

Manav Seksaria with,
Sourav Dutta and Anubhab Rudra

IIT Madras

November 14, 2024

Table of Contents

Qudits: QHMs

There are two schools of thought:

Those who *think* the universe is made up Quantum Harmonic Oscillators

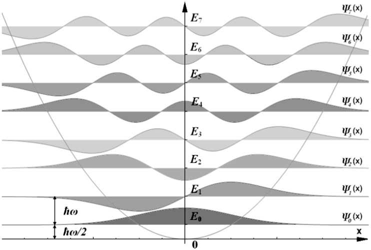

Figure: Quantum Harmonic Oscillator

And those who don't think at all.

Qudits: QHMs

$$\psi_n(x) = \frac{C}{\sqrt{2^n\, n!}} e^{-\frac{\alpha x^2}{2}} H_n\left(\sqrt{\alpha}x\right)$$

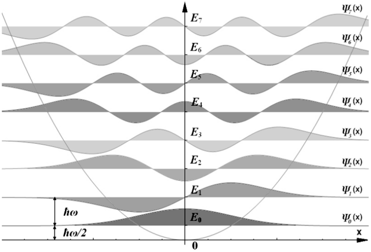

Figure: QHM

Therefore for qubits:

$$\psi_0 = Ce^{-\alpha x^2}$$

$$\psi_1 = Ce^{-\alpha x^2}\sqrt{2\alpha}x$$

and $E_0 = \frac{\hbar\omega}{2}$, $E_1 = \frac{3\hbar\omega}{2}$

But higher dimensions are possible even if they're difficult to physically realize for now.

Qudits: Anharmonicity

We need anharmonicity to be able to distinguish b/w states, because energy levels of a QHM are equidistant.

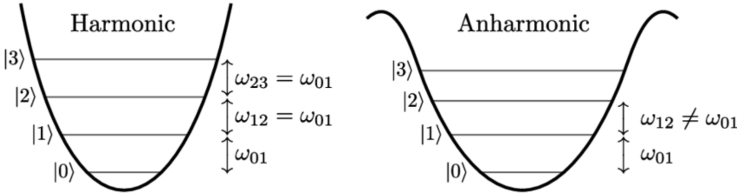

Figure: Quantum Anharmonic Oscillator

For superconducting qubits (**transmon**), this is achieved by using Josephson Junctions. In an LC Oscillator, we replace the linear inductance with a non-linear one.

Qudits $\rightarrow$ GAD

Why is physical realization of qudits difficult? $\exists$ several forms of noise, most prominent being dephasing and relaxation.
Amplitude Damping:

$$|1\rangle \rightarrow e^{-\frac{t}{T_1}}|1\rangle$$

With some probability, the state $|1\rangle$ will decay to $|0\rangle$ with time constant T_1. Let's call this probability γ, such that $|1\rangle \rightarrow \sqrt{1-\gamma}|1\rangle + \sqrt{\gamma}|0\rangle$.

Qudits $\rightarrow$ GAD

In general, $|2\rangle \rightarrow |1\rangle$, and, $|1\rangle \rightarrow |0\rangle$. But what may also happen is $|2\rangle \rightarrow |0\rangle$ directly with some probability $\gamma_2 \propto \gamma^2$ and so on upto γ^d.

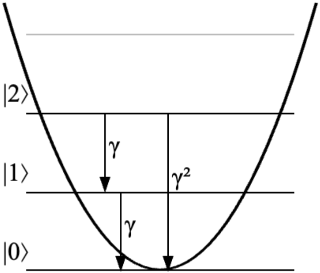

Figure: Extended Amplitude Damping

For our considerations, we'll consider the case where $\gamma_2 = \gamma^2$ since we're worried only about order of noise and not the exact values.

Qudits $\rightarrow$ GAD

In the other direction, $|0\rangle \rightarrow |1\rangle$ with probability γ and $|1\rangle \rightarrow |2\rangle$ with probability γ^2.

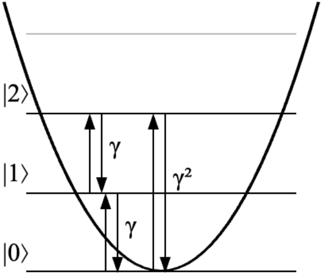

Figure: Generalized Amplitude Damping

This is the generalized amplitude damping channel. We'll consider this as our noise model for the rest of the presentation.

GAD

It is worth noting there is some nuance in how the probabilities for GAD are reported,
For decay, it is always $\mathbb{O}(\gamma^n)$, but for excitation, it has a prefix p, which is a function of temperature T

$$\begin{cases} T = 0 \rightarrow p = 0 \\ T = \infty \rightarrow p = \gamma \end{cases}$$

Therefore, it is in fact common to report p as a fraction of γ most commonly $\frac{\gamma}{10}$ or $\frac{3\gamma}{10}$

QEC: Modelling Noise

Noise is modelled via an error channel $\mathcal{E}$ acting on the state ρ of the system

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger, \ \sum_i E_i^\dagger E_i = I$$

The canonical qubit Kraus operators for relaxation and excitation respectively are:

$$A_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix},$$
$$R_0 = \begin{bmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{bmatrix}$$

QEC: Correctability

In general noise is correctable only if it satisfies two conditions:

- $\langle i|E_a^\dagger E_b|j\rangle = 0$ if $i \neq j$. Orthogonality condition: no two subspaces have any intersection.
- $\langle i|E_a^\dagger E_b|i\rangle =$ const, Deformity condition: the subspaces are NOT deformed by the noise.

Both of these combine to give us the Knill-Laflamme conditions for correctability.

$$\langle i_L|E_a^\dagger E_b|j_L\rangle = \lambda_{ab}\delta_{ij}$$

QEC: Approximate QEC

In general exact QEC is difficult to achieve. We get around this by using approximate QEC via the Beny-Oreshkov condition:

$$\langle i_L | E_a^\dagger E_b | j_L \rangle = \lambda_{ab} \delta_{ij} + \langle i_L | B_{ab} | j_L \rangle$$

where B_{ab} is the error term to allow for approximate QEC.

QEC: Overhead

In general error correction has two extra steps:

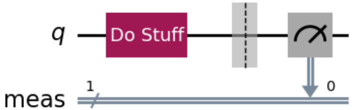

Figure: Before Error Correction

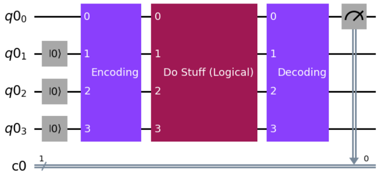

Figure: After Error Correction we add extra steps before and after

QEC: Encoding

We tend to encode data from lower to higher dimensions for redundancy, ex. the trivial code is $|000\rangle \rightarrow |0_L\rangle$ and $|111\rangle \rightarrow |1_L\rangle$. Some popular codes are the Shor code, the CSS code, stabilizer codes etc.
Generally any code is:

$$|0_L\rangle = \sum c_i \bigotimes_{i=1}^{d} |i\rangle$$

where c_i are some coefficients. Or even more generally,

$$|0_L\rangle = T|\psi_0\rangle, |1_L\rangle = T|\psi_1\rangle, \ldots$$

where T is a rectangular transformation matrix applied to some state $|\psi_0\rangle$ for ψ_i in the physical space.

Finding T

We can now construct a system such that:

$$\langle i|T^\dagger E_a^\dagger E_b T|j\rangle = \lambda_{ab}\delta_{ij} + \langle i_L|B_{ab}|j_L\rangle$$

with three weights in the cost function:

- $||TT^T - \text{diag}(TT^T)||_F \to 0$, for orthogonality $\because$ TT^T is diagonal
- $\sum_i (1 - ||T_i||)^2 \to 0$, to ensure norm $= 1$, such that T_i are the rows of T
- $||B_{ab}||_F \to 0$, Error matrix tend to zero

Finding T: Algorithm

For $B_{ab} \in \mathbb{R}^{l^k \times l^k}$:

Algorithm 1 Finding Loss

$L_{\text{AQEC}} \leftarrow 0$
$L_{\text{ortho}} \leftarrow ||TT^T - \operatorname{diag} TT^T||_F$
$L_{\text{norm}} \leftarrow \sum_i (1 - ||T[i]||_2)^2$
for E_a, E_b in $\{E_i\}$ **do**
    for i, j in basis of dim l^k **do**
        $B_{ab}[j, i] = \langle j | T^\dagger E_a^\dagger E_b T | i \rangle$
    end for
    $L_{\text{AQEC}} \leftarrow L_{\text{AQEC}} + ||B_{ab} - \operatorname{Tr}(B_{ab}) \cdot \mathbb{I}||_F$
end for
return $L_{\text{AQEC}} + L_{\text{norm}} + L_{\text{ortho}}$

Choosing T

We choose initial T to minimise optimization time. Let T_i be a Legendre polynomial such that, for $P_0(x) = 1, P_1(x) = x$:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$

For reference, the next few Legendre polynomials are:

$$P_2(x) = \frac{1}{2}(3x^2 - 1)$$
$$P_3(x) = \frac{1}{2}(5x^3 - 3x)$$
$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

Choosing T

The advantage then is that all rows are orthogonal:

$$\int_{-1}^{1} P_m(x)P_n(x)\,dx = 0 \quad \forall n \neq m$$

We trivially normalise maintaining orthogonality, with $k_m = \int_{-1}^{1} P_m(x)^2\,dx$:

$$\int_{-1}^{1} \frac{P_m(x)}{k_m}\frac{P_n(x)}{k_n}\,dx = \frac{2}{2n+1}\frac{\delta_{mn}}{k_m k_n}$$

Note: We choose Legendre's polynomials because they're orthogonal in $[-1, 1]$, unlike Hermite's in $(-\infty, \infty)$. It also has a constant weight unlike Chebyshev's.

Running the Gradient Descent

Having run the gradient descent, we get the following code-words:

$$\begin{aligned}|0_L\rangle = {} & 0.077|0000\rangle - 0.192|0001\rangle - 0.108|0010\rangle + 0.326|0011\rangle \\ & - 0.087|0100\rangle + 0.582|0101\rangle + 0.101|0110\rangle + 0.096|0111\rangle \\ & - 0.250|1000\rangle + 0.229|1001\rangle - 0.032|1010\rangle + 0.300|1011\rangle \\ & + 0.334|1100\rangle + 0.243|1101\rangle + 0.301|1110\rangle + 0.087|1111\rangle\end{aligned}$$

$$\begin{aligned}|1_L\rangle = {} & 0.091|0000\rangle + 0.084|0001\rangle - 0.153|0010\rangle + 0.379|0011\rangle \\ & + 0.035|0100\rangle - 0.666|0101\rangle + 0.161|0110\rangle - 0.157|0111\rangle \\ & - 0.187|1000\rangle - 0.053|1001\rangle + 0.013|1010\rangle + 0.191|1011\rangle \\ & + 0.207|1100\rangle - 0.147|1101\rangle + 0.413|1110\rangle + 0.108|1111\rangle\end{aligned}$$

Comparison

We compare our **four** qubit code against the standard benchmark of **five** qubit $[[5,1,3]]$ code, via Petz Recovery at $\gamma = 0.1$.

Stat	$[[5,1]]$	Leung	Ours
$p = \gamma/10$	0.863	0.825	0.881
$p = 3\gamma/10$	0.901	0.831	0.887
$p = \gamma$	0.937	0.850	0.896

Table: Entanglement Fidelity

$$F = \frac{1}{(\dim C)^2} \sum_k \sum_l |Tr(R_l A_k)_{|C}|^2$$

We consistently outperform the benchmark Leung code, and even do better than $[[5,1]]$ for $\gamma/10$

QEC: Recovery

In general there are three ways to do recovery:

- Simply project back $|0_L\rangle \to |0_L\rangle\langle 0_L|$
- Use Petz Recovery: $R = \sum_k \sum_l \mathrm{Tr}(R_l A_k)|k\rangle\langle l|$
- Construct a recovery operator $|0_L\rangle \to |0_L\rangle\langle 0'_L|$

QEC: Parametrised Form

Consider code word $|0_L\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$, then the trivial projected back recovery is:

$$|0_L\rangle \to |0_L\rangle \left(\frac{1}{\sqrt{2}} \langle 0000| + \frac{1}{\sqrt{2}} \langle 1111| \right).$$

We, can however, run a gradient descent and find a more optimal recovery:

$$|0_L\rangle \to |0_L\rangle (\alpha \langle 0000| + \beta \langle 1111|).$$

QEC: Parametrised Form (1st Order)

The first would then require a first order damping applied to each qubit. $\{A_{0001}, A_{0010}, A_{0100}, A_{1000}\}$. Which as an example, for A_{0001} would be:

$$A_{0001}|1_L\rangle = A_{0001}|1111\rangle \rightarrow |1110\rangle$$

which would then mean we have to construct a recovery from the new state $|1110\rangle$ to 4-GHZ.

QEC: Recovery Loss

Algorithm 2 Loss for Recovery of order p

$R_k \leftarrow []$
$L_{\text{ortho}} \leftarrow 0$
$L_{\text{norm}} \leftarrow 0$
for $\{\psi_i\} \, \forall \, A_k$ **do**
  $\{|\phi_i\rangle\} \leftarrow \text{reweight}(\{|A_k \psi_i\rangle\})$
  $L_{\text{ortho}} \leftarrow L_{\text{ortho}} + \sum_{i,j} \langle \phi_i | \phi_j \rangle^2$
  $L_{\text{norm}} \leftarrow L_{\text{norm}} + \sum_i (1 - ||\phi_i||)^2$
  $R_k \leftarrow \sum_i |\psi_i\rangle\langle\phi_i|$
end for
$L_{\text{fidelity}} \leftarrow \sum_k \sum_l |Tr(R_l A_k)_{|C}|^2$
return $L_{\text{ortho}} + L_{\text{norm}} - L_{\text{fidelity}}$

For Leung code, we are able to replicate the known optimal recovery.

QEC: Comparison

We can now compare our code against our recovery or the petz recovery

Stat	Petz Recovery	Ours
$p = \gamma/10$	0.9996	0.9997
$p = 3\gamma/10$	0.9992	0.9995
$p = \gamma$	0.9980	0.9987

Table: Full Channel Entanglement Fidelity, both for our code

It is worth noting, we have yet done only 0th order recovery, and the Petz recovery is near optimal for all orders.

Combining

We now have a function which can generate a code and one for recovery. We can do one of two approaches:

- Run Separately: Code $\rightarrow$ Recovery $\rightarrow$ Better Code $\rightarrow$ Better Recovery $\rightarrow \dots$
- Run Together: GAN

Considering that the separately is computationally cheaper and can get 0.999s of reliability we will use that. Despite the fact that the GAN is garunteed to be more optimal.

GAN: Optimality

let $f(x) \in V, g(y) \in W$ be optimisation functions. To ensure optimality of GAN, we need to prove:

$$vw \geq v + w, \text{ where } v = |V|, w = |W|$$

WLOG, let $w = v + k$, then:

$$\begin{aligned} v + v + k - v(v+k) &\leq 0 \\ \implies v^2 - v(k+2) - k &\geq 0 \\ \implies v = \frac{k + 2 \pm \sqrt{k^2 + 8k + 4}}{2} \end{aligned}$$

Therefore, for all natural values of v, w, and $vw \geq v + w$.

Next Steps

We have a working code for GAD on qudits, and a recovery operator. We can now move on to:

- Calculate 4 qubit for 1st and 2nd order recovery
- Run for 5 qubits because we expect to be able to correct GAD
- For more qubits we'll have enough data to find thresholds
- Qubit-Qutrit system