

**UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS**

Alexandre de Taunay Voloch

**Caracterização e Supressão de Ruído Ambiental em Sistemas
Quânticos Usando Redes Neurais**

São Carlos

2025

Alexandre de Taunay Voloch

Caracterização e Supressão de Ruído Ambiental em Sistemas Quânticos Usando Redes Neurais

Trabalho de conclusão de curso apresentado ao Programa de Graduação em Física do Instituto de Física de São Carlos, da Universidade de São Paulo, para a obtenção do título de Bacharel em Física.

Orientador: Prof. Dr. Reginaldo de Jesus Napolitano

Versão original

**São Carlos
2025**

RESUMO

VOLOCH, A. **Caracterização e Supressão de Ruído Ambiental em Sistemas Quânticos Usando Redes Neurais**. 2025. 23 p. Monografia (Trabalho de Conclusão de Curso) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2025.

Este trabalho, baseado no projeto de iniciação científica 2024/14189-2, financiado pela FAPESP, tem como objetivo desenvolver métodos que combinam técnicas numéricas avançadas, teoria de controle quântico e aprendizado de máquina para caracterizar e mitigar o ruído ambiental em simulações de sistemas quânticos de um qubit do tipo transmon truncado em três níveis. Utilizando um modelo simplificado de vazamento (leakage) entre o nível lógico $|1\rangle$ e o indesejado $|2\rangle$, desenvolvemos um protocolo utilizando medições de pulsos específicos, simulados no sistema Qiskit da IBM, em conjunto com uma rede neural para caracterizar com boa acurácia as frequências de ruído do sistema. A partir disso, aplicamos o desacoplamento dinâmico contínuo generalizado (GCDD) para efetivamente mitigar o ruído unitário nesse sistema, possibilitando aumento significativo da fidelidade das operações quânticas, contribuindo de maneira relevante para avanços futuros que ultrapassem as limitações atuais da era NISQ (Noisy Intermediate-Scale Quantum).

Palavras-chave: Computação quântica. Mitigação de ruído. Rede neural. Desacoplamento dinâmico. Teoria de controle quântico. Sistemas quânticos.

1 INTRODUÇÃO

Desde o desenvolvimento da mecânica quântica na metade do século XX, matemáticos, físicos e cientistas da computação teorizaram sobre a possibilidade de construir computadores quânticos, que trariam avanços significativos em diversas áreas da ciência e do desenvolvimento humano, como medicina, química, criptografia, entre muitas outras (1–6). Utilizando efeitos decorrentes da mecânica quântica como superposição e emaranhamento, esses equipamentos possibilitam resolver certos tipos de problema drasticamente mais rápido do que com computadores convencionais (chamados computadores clássicos) (7), com consequências economicamente significativas.

No entanto, a computação quântica encontra-se atualmente na chamada era NISQ (*Noisy Intermediate-Scale Quantum*) (8), caracterizada por dispositivos ruidosos de tamanho intermediário, úteis para exploração física e algoritmos de pequena escala, mas ainda aquém da computação tolerante a falhas e efetivamente útil. Isto significa que, apesar de dispositivos com centenas de bits quânticos (qubits) haverem sido desenvolvidos por empresas como IBM, Quantinuum e Microsoft (9–12), estes ainda não apresentam escala e estabilidade suficientes para execução de algoritmos e programas que apresentem vantagem efetiva sobre os computadores existentes, não quânticos (7), vantagem denominada vantagem quântica ou *quantum advantage* (13).

Em 2025, o Prêmio Nobel de Física reconheceu experimentos pioneiros com circuitos supercondutores macroscópicos que demonstraram tunelamento quântico e níveis de energia quantizados em sistemas "grandes o suficiente para caber na mão" (14), e a nota científica oficial cita explicitamente como decorrente dessa pesquisa a plataforma de qubits supercondutores *transmon* (15), amplamente utilizada na computação quântica atual. Neste trabalho, utilizamos a física do *transmon* como base para nossos modelos e simulações.

Uma das principais fontes da instabilidade inerente aos computadores quânticos atuais é o ruído. Diferentemente da computação clássica, onde estados são relativamente estáveis e independentes do ambiente, computadores quânticos são extremamente sensíveis a variações externas e internas, denominadas de "ruído". Por exemplo, computadores quânticos requerem operações a temperaturas extremamente baixas, da ordem de 10 – 20 mK (16). Os efeitos principais do ruído são (17):

- Mudança, ao longo do tempo, do estado configurado inicialmente, levando a perda de informação
- Destruição de coerência e superposição em um canal de comunicação

- Uma porta lógica (operação) implementada de forma imperfeita, gerando resultados inesperados

O ruído pode ser unitário ou não-unitário, e proveniente de diversas fontes no sistema ou no ambiente, mas resulta, sempre, em uma perturbação na evolução esperada do sistema, dificultando ou impossibilitando a realização de medidas com acurácia adequada. Portanto, o tratamento e a mitigação do ruído são de importância significativa para o desenvolvimento de computadores quânticos robustos e confiáveis. Existem várias técnicas de tratamento de ruído, como códigos de correção de erro (18) e arquiteturas com qubits lógicos, compostos por mais do que um qubit físico. Concomitantemente, há pesquisa em mitigação (redução) do ruído em si, com técnicas como desacoplamento dinâmico (19) e otimização de pulsos (20), área de pesquisa denominada teoria de controle quântico.

O nosso grupo de pesquisa, liderado por R. J. Napolitano, vem desenvolvendo técnicas avançadas de desacoplamento dinâmico contínuo generalizado (GCDD) (21, 22). Em relação à lista acima, o GCDD atua para mitigar o primeiro tipo de ruído, o de evolução indesejada do sistema ao longo do tempo, especificamente quando um está "ocioso", isto é, passa bastante tempo sem que uma operação seja realizada nele. No formalismo da mecânica quântica, diz-se que há um operador de evolução $U(t)$ (que satisfaz a equação de Schrödinger $i\hbar \frac{dU(t)}{dt} = H(t)U(t)$) regendo o sistema, e que gostaríamos de modificar a hamiltoniana $H(t)$, através de pulsos de controle, para que $U(t) = I$, onde I é a identidade, ou seja, queremos que o sistema permaneça constante ao longo do tempo.

Um tipo de ruído é o *leakage* (vazamento) (23), que existe em sistemas onde existem níveis de energia além do subespaço lógico do qubit (isto é, $\{|0\rangle, |1\rangle\}$). Em um bit clássico (por exemplo, em um computador convencional), existem apenas dois estados possíveis: 0 e 1, correspondendo ao transistor estar "ligado" ou "desligado". Entretanto, muitas plataformas quânticas, como a supracitada de qubits *transmon*, possuem mais do que dois níveis possíveis de energia, resultando na possibilidade de o qubit encontrar-se em um estado excitado acima do subespaço lógico, como por exemplo o estado $|2\rangle$, acarretando resultados inesperados e perda de acurácia, e trazendo a necessidade de técnicas de mitigação do *leakage*.

Em sistemas quânticos reais, o *leakage* é consequência da dependência temporal da amplitude do pulso de controle, que pode excitar o qubit ao nível do segundo estado excitado, que difere do primeiro estado excitado por uma diferença de energia $\Delta E_{1,2}$ menor do que a diferença de energia entre o primeiro estado excitado e o estado fundamental $\Delta E_{0,1}$. Neste projeto, para fins pedagógicos, escolhemos utilizar um modelo de *leakage* estático, isto é, sem dependência temporal. Este modelo equivaleria ao *leakage* que ocorre durante a execução de um mesmo pulso de controle ao longo de várias rodadas de um circuito.

Neste trabalho, utilizamos este modelo de *leakage* estático, em um sistema *transmon*

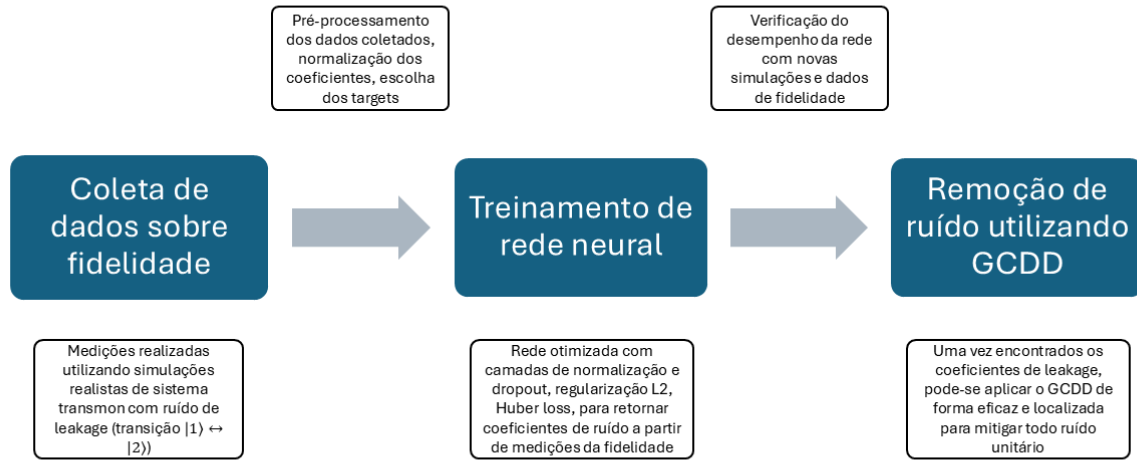


Figura 1 – Esquematização do planejamento do trabalho.

Fonte: Elaborada pelo autor.

de um qubit truncado a três níveis, como base para desenvolver um protocolo de mitigação de ruído, que possibilita obter com alta acurácia as frequências do ruído, utilizando medições de estados preparados com pulsos específicos, em conjunto com uma rede neural. Uma vez obtidas as frequências de ruído, aplicamos o GCDD para mitigar o ruído e manter alta fidelidade mesmo com tempo alto de ociosidade. Um pipeline de desenvolvimento deste trabalho pode ser visualizado na figura 1.

Escolhemos uma rede neural por duas razões. Primeiramente, redes neurais são aproximadores universais, com precisão arbitrária, de qualquer função contínua (24). Portanto, assumindo que existe uma relação entre os parâmetros de entrada (nesse caso, as fidelidades resultantes de aplicar certos pulsos de controle) e os parâmetros de saída (as frequências do ruído), uma rede neural de tamanho suficiente conseguirá aproximar essa relação e, portanto, prever os parâmetros de ruído a partir apenas das medições feitas. Segundo, mesmo em situações físicas onde haja aproximação com solução analítica, aplicar redes neurais permite, em muitos casos, generalização para além de casos simplificados. Para este trabalho, por exemplo, é possível chegar em soluções analíticas, porém mesmo ao adicionarmos outras frequências de ruído ao sistema, a rede neural ainda consegue prever com boa acurácia as frequências de *leakage*.

2 MATERIAIS E MÉTODOS

Este capítulo descreve, de forma reprodutível, a arquitetura de simulação utilizada, a modelagem físico-matemática adotada para o *transmon* truncado, a definição dos pulsos de controle e do conjunto de medidas, o protocolo de geração de dados e tratamento dos mesmos, e a arquitetura de rede neural utilizada para a inferência dos parâmetros de *leakage*.

2.1 Ambiente de simulações

Todas as simulações foram feitas utilizando a linguagem de programação *Python* e o ecossistema *Qiskit*. O *Qiskit* (25) é uma família de bibliotecas para computação quântica de código livre, desenvolvidas pela IBM, com o intuito de ser uma plataforma unificada para geração de circuitos, compilação para pulsos, e execução, tanto em computadores quânticos reais (via *IBM Cloud Runtime*) como em plataformas simuladas utilizando o *Qiskit Dynamics* (26).

2.1.1 Digressão sobre o ambiente *Qiskit*

Este projeto é baseado em um projeto de iniciação científica, financiado pela FAPESP, cujo âmbito original incluía realizar medições reais de resultados de execução de pulsos, nos computadores da IBM. Porém, no final de 2024, a IBM anunciou a depreciação do pacote *Qiskit Pulse*, sendo que a partir do início de 2025 não seria mais possível executar pulsos de controle nos sistemas próprios da empresa (vide (27)). Para dar continuidade ao projeto, escolhemos utilizar o pacote de simulações *Qiskit Dynamics*.

Infelizmente, os pacotes de simulação do ecossistema *Qiskit* têm tido baixa manutenção. O próprio *Dynamics* recebeu sua última atualização (em 07/10/2025) com a informação de que a biblioteca não seria mais mantida ativamente. Para que as simulações descritas no decorrer desta seção funcionem, recomendamos a utilização das versões dos pacotes de acordo com a tabela 1. Porém, em última instância, recomendamos que pesquisadores na área de teoria de

Tabela 1 – Versões dos pacotes utilizados

Nome do pacote	Versão utilizada
<code>qiskit</code>	$< 1.3.0$
<code>qiskit_aer</code>	$\geq 0.15.0$
<code>qiskit_algorithms</code>	$\geq 0.3.0$
<code>qiskit_dynamics</code>	$= 0.5.1$
<code>qiskit_experiments</code>	$\geq 0.7.0$
<code>qiskit_ibm_runtime</code>	$\approx 0.29.0$

Fonte: Elaborada pelo autor.

controle quântico se afastem do ambiente *Qiskit* em prol de outros ecossistemas melhor mantidos, como o *QuTiP* (28) para simulações e o *Amazon Braket* (29) para execução de pulsos.

2.1.2 Modelo utilizado

Para os propósitos deste trabalho, escolhemos nos basear em um dos modelos já inclusos no *Qiskit*, que é fornecido com base em um chip existente da IBM, neste caso chamado **FakeArmonk**, de 1 qubit. Os objetos de modelo no *Qiskit* possuem uma descrição interna da dinâmica utilizada. No caso deste chip, a hamiltoniana do modelo fornecido é a de um oscilador de Duffing, do formato*

$$\mathcal{H}/\hbar = \omega_q N - \frac{\Delta}{2}(N^2 - N) \quad (2.1)$$

Onde ω_q, Δ são frequências positivas (respectivamente, a frequência base do qubit e termo de anarmonicidade), da ordem de GHz, e $N = a^\dagger a$ é o operador número. No decorrer desse trabalho, não iremos utilizar o formalismo de levantamento e abaixamento mas sim modelar o qutrit a partir das matrizes de Gell-Mann. Portanto, a equação 2.1 é dada apenas como referência ao modelo interno realizado pela IBM.

Esta hamiltoniana representa um oscilador de Duffing, isto é, um oscilador levemente anarmônico, e é a modelagem física padrão para a junção de Josephson presente no qubit supercondutor (30). Os níveis de energia são truncados no terceiro nível, isto é, o sistema possui como base os níveis $\{|0\rangle, |1\rangle, |2\rangle\}$, sendo os primeiros dois chamados de "base lógica" ou "base qubit". O sistema como um todo é um qutrit, isto é, um qubit com três níveis possíveis.

Para utilizarmos a equação 2.1 no nosso sistema, precisamos transformar os operadores em matrizes, que serão aplicados nos vetores de estado em cada iteração da simulação, para evoluir o sistema de acordo com a equação de Schrödinger. Uma vez que $N|n\rangle = n|n\rangle$, é trivial de encontrar a matriz do operador N , chegando em

$$\mathcal{H}/\hbar = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \omega_q & 0 \\ 0 & 0 & 2\omega_q - \Delta \end{pmatrix} \quad (2.2)$$

Percebe-se que a eq. 2.2 contém apenas termos diagonais, o que significa que o modelo fornecido pela própria IBM é apenas uma aproximação da hamiltoniana verdadeira do sistema, que deve conter termos não diagonais, como o do *leakage*. Para este projeto, utilizamos como inspiração a equação fornecida pela IBM, porém realizamos a modelagem mais generalizada possível para o qutrit, através das matrizes de Gell-Mann.

* A equação e propriedades desse sistema específico podem ser acessadas através do objeto `fake_provider.FakeArmonk`, do pacote `qiskit_ibm_runtime`, através dos métodos `configuration()` e `properties()`.

Assim como as matrizes de Pauli são uma base completa para matrizes hermitianas 2x2 (e são geradoras de $\mathfrak{su}(2)$), as matrizes de Gell-Mann são uma base completa para matrizes hermitianas 3x3 e geradoras de $\mathfrak{su}(3)$. Portanto, é natural expressarmos uma hamiltoniana generalizada como uma combinação linear das matrizes de Gell-Mann:

$$H = \hbar \sum_{i=0}^8 \omega_i \lambda_i \quad (2.3)$$

onde λ_i é a i -ésima matriz de Gell-Mann e ω_i são as frequências associadas.

De relevância para este trabalho são as matrizes λ_6 e λ_7 , que são análogas a σ_x e σ_y no subespaço de *leakage* $\{|1\rangle, |2\rangle\}$:

$$\lambda_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}; \quad \lambda_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix} \quad (2.4)$$

sendo as demais matrizes de Gell-Mann omitidas por razões de espaço. Portanto, o hamiltoniano para um problema de ruído *leakage* é

$$H_{drift} = \hbar(\omega_6 \lambda_6 + \omega_7 \lambda_7) = H_{drift} = \hbar \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \omega_6 - i\omega_7 \\ 0 & \omega_6 + i\omega_7 & 0 \end{pmatrix} \quad (2.5)$$

Finalmente, desejamos que nossa análise seja a mais generalizada possível. Portanto, incorporamos uma constante α que é a ordem de grandeza relativa das outras frequências, em relação às frequências de *leakage*, de tal forma que o hamiltoniano que representa o modelo simulado seja

$$H_{drift} = \hbar \left[\alpha \left(\sum_{i \neq (6,7)} \omega_i \lambda_i \right) + \omega_6 \lambda_6 + \omega_7 \lambda_7 \right] \quad (2.6)$$

Note que a equação 2.6 se reduz à 2.5 quando $\alpha = 0$. Portanto, α representa justamente a ordem de grandeza relativa das outras frequências, que irão interferir na dinâmica "esperada" do sistema (de apenas *leakage*.)

2.1.3 Simulação

No ambiente de simulação, para simplificar as representações numéricas, colocamos como base das nossas unidades o tempo total do experimento $\tau = 4\mu s = 4(1 \times 10^{-6}s)$ e estabelecemos, com base na literatura (31), o valor máximo das frequências de *leakage* como $1\text{MHz} = 2\pi(1 \times 10^6 \text{ rad s}^{-1})$.

Para a simulação dos sistemas quânticos, utilizamos o objeto **Solver** da biblioteca *Qiskit Dynamics*. Para simular os pulsos enviados ao qubit (vide eqs. 2.18, 2.23) utilizamos o objeto

`Signal` da mesma biblioteca. Para otimizar a velocidade das simulações, foi utilizada a biblioteca `jax`, que permite compilação de funções matemáticas no *Python*, melhorando substancialmente a velocidade.

2.2 Desacoplamento dinâmico

2.2.1 Intuição

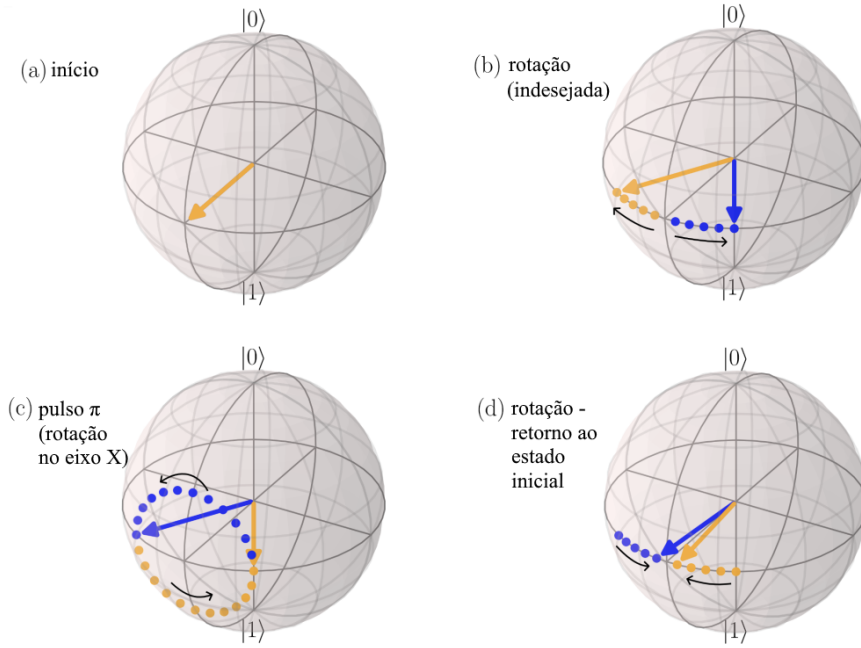


Figura 2 – Visualização do desacoplamento dinâmico simples por meio do pulso π_x .
Fonte: Adaptado da original, da equipe Amazon Braket (32).

O desacoplamento dinâmico (DD) é uma técnica de mitigação de qualquer ruído unitário, que seja quase-estático ou estático em relação ao tempo do desacoplamento t_0 . Aqui, vamos tentar fornecer uma intuição sobre o que o DD "faz", e na seção seguinte vamos destrinchar a formulação geral do desacoplamento dinâmico contínuo (GCDD) realizada pelo nosso grupo. Essa seção baseia-se em uma postagem didática realizada pela equipe do Amazon Braket (32).

Um qubit em um estado arbitrário pode ser modelado como um vetor na esfera de Bloch. Processos físicos que interferem no estado original do qubit, desde que unitários, efetuam rotações desse vetor ao redor da esfera. Vamos assumir, neste exemplo, que essas rotações são de duração suficiente que, em um tempo determinado t_0 , elas possam ser aproximadas por constantes. A estratégia básica do desacoplamento dinâmico é de desfazer essas rotações, por meio de um pulso que rotacione o qubit em um eixo perpendicular ao do ruído, de tal forma que a própria rotação gerada pelo ruído agora faça o qubit retornar à posição onde estava (só que do outro lado da esfera), e depois rotacionar o qubit de volta, para que ele retorne ao seu estado inicial.

Vamos supor que um qubit possa se rotacionar de duas formas, que vamos representar pelo estado $|\odot\rangle$, para uma rotação anti-horária, e $|\oslash\rangle$, para uma rotação horária, ambos em relação ao eixo Z.

Nosso qubit inicia no estado $|\odot\rangle$. Após um certo tempo t_0 , o qubit, inicialmente centrado na esfera de Bloch, sofrerá uma mudança de fase, isto é, ele rotacionará ao redor do eixo Z. Uma forma simples de DD é de enviar um pulso π_x ao qubit, isto é, efetuar uma rotação de 180° ao redor do eixo X, de tal forma que o seu estado de rotação passe a ser o $|\oslash\rangle$. Então, após esperar novamente por um tempo t_0 , o qubit acabará na mesma posição que ele estava no início da dinâmica. Se realizarmos nossas medidas no qubit, apenas em instantes de tempo múltiplos de t_0 , então não perceberemos que há um ruído afetando ele. A dinâmica desse exemplo pode ser visualizada na figura 2.

Com essa intuição de como o DD (neste caso, o discreto) age no qubit, partiremos para a formulação geral e contínua.

2.2.2 Formulação geral do GCDD

Nesta seção, vamos mostrar como efetuar uma operação arbitrária em um qudit sem a interferência significativa de ruído. Aqui, os detalhes da implementação física do processo serão deixados de lado, e iremos apenas demonstrar como modificar a hamiltoniana para alcançar o efeito desejado.

Seja $H_G(t)$ a hamiltoniana desejada (ideal) e $H_c(t)$ o pulso de controle que irá mitigar o ruído até primeira ordem. A hamiltoniana total, com o ruído, é dada por:

$$H(t) = (H_G(t) + H_c(t)) \otimes I_E + I_Q \otimes H_E + H_{int} \quad (2.7)$$

Na equação acima, I_E e I_Q são, respectivamente, os operadores de identidade do subespaço do ambiente (externo ao qudit), e do qudit em si. H_E é uma hamiltoniana qualquer (relacionada à evolução do ambiente) e H_{int} é a interação entre o qudit e o ambiente, isto é, o ruído.

Uma das características do controle é que ele deve ser periódico, para que possamos realizar a expansão de Magnus. Sendo τ o tempo total de evolução (pode representar o tempo de ociosidade do qudit, ou o tempo de operação de uma porta lógica), definimos $t_0 = \frac{\tau}{n}$ onde $n \in \mathbb{N}$ é um número inteiro positivo. Veremos que quanto maior o valor de n , mais efetiva será a aniquilação do ruído.

O ruído será mitigado na representação de interação da hamiltoniana de controle. Portanto, definimos o operador evolução $U_c(t)$ da hamiltoniana $H_c(t)$, regido pela equação de Schrödinger:

$$i\hbar\dot{U}_c(t) = H_c(t)U_c(t) \quad (2.8)$$

Realizando a mudança de representação na eq. 2.7, temos

$$H_I(t) = U_c^\dagger(t) H_G(t) U_c(t) + I_Q \otimes H_E + (U_c^\dagger(t) \otimes I_E) H_{int} (U_c(t) \otimes I_E)$$

Queremos encontrar $H_c(t)$ tal que H'_{int} , isto é, o ruído na representação de interação, seja proporcional à identidade \mathbf{I} .

Se o controle é periódico com período t_0 e $\tau = nt_0$ é a duração total do experimento, então, em 1ª ordem da expansão de Magnus,

$$U_I(\tau) \approx \exp\left(-\frac{i}{\hbar}\tau \langle H_I \rangle_\tau\right), \quad (2.9)$$

onde $\langle A \rangle_t = \frac{1}{t} \int_0^t A dt$. Portanto, a hamiltoniana efetiva no subespaço do qudit é

$$\begin{aligned} H_I(t) &\approx \langle H_I(t) \rangle_\tau = \langle U_c^\dagger(t) H_G(t) U_c(t) \rangle_\tau + \langle U_c^\dagger(t) H_{int} U_c(t) \rangle_\tau \\ &= \langle U_c^\dagger(t) H_G(t) U_c(t) \rangle_\tau + \langle U_c^\dagger(t) H_{int} U_c(t) \rangle_{t_0}, \end{aligned} \quad (2.10)$$

uma vez que $U_c(t)$ é periódico em t_0 .

Para construir nosso pulso de controle, vamos definir

$$H_L |k\rangle \equiv k\hbar\omega_d |k\rangle, \quad H_F |\psi_n\rangle \equiv n\hbar\omega_0 |\psi_n\rangle, \quad (2.11)$$

onde $\omega_0 = \frac{2\pi}{t_0} = \frac{2\pi n}{\tau}$ e $\omega_d = d\omega_0$ (onde d é o número de níveis de energia do qudit - para um qubit, $d = 2$, para um qutrit, $d = 3$, etc.) $\{|k\rangle\}$ é a base lógica do qudit, e $\{|\psi_n\rangle\}$ é a transformada de Fourier quântica desta, dada por

$$|\psi_n\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} \exp\left(\frac{2\pi i}{d} jn\right) |j\rangle \quad \left(\Leftrightarrow |j\rangle = \frac{1}{\sqrt{d}} \sum_{m=0}^{d-1} \exp\left(-\frac{2\pi i}{d} jm\right) |\psi_m\rangle\right) \quad (2.12)$$

Com essas definições, vamos escolher construir o operador evolução do controle $U_c(t)$ da seguinte forma:

$$U_c(t) = e^{-\frac{i}{\hbar} H_L t} e^{-\frac{i}{\hbar} H_F t} \equiv U_L(t) U_F(t), \quad (2.13)$$

e encontramos $H_c(t)$ invertendo a eq. 2.8:

$$H_c(t) = i\hbar \dot{U}_c(t) U_c^\dagger(t) = (H_L U_L U_F + U_L H_F U_F) (U_L U_F)^\dagger = H_L + U_L(t) H_F U_L^\dagger(t) \quad (2.14)$$

Para provar que essa hamiltoniana de controle remove o ruído, vamos construir o nosso operador de interação arbitrário como (sendo c_{rs} constantes quaisquer)

$$H_{int} = \sum_{r=0}^{d-1} \sum_{s=0}^{d-1} c_{rs} |r\rangle \langle s| \quad (2.15)$$

Vamos encontrar a média, na representação de interação, dos elementos $|p\rangle\langle q|$. Para facilitar a notação vamos escrever $\langle (U_c^\dagger(t) |p\rangle\langle q| U_c(t)) \rangle_{t_0} \equiv M_{pq}$:

$$\begin{aligned}
M_{pq} &= \int_0^{t_0} dt U_c^\dagger(t) |p\rangle\langle q| U_c(t) = \int_0^{t_0} dt U_F^\dagger(t) U_L^\dagger(t) |p\rangle\langle q| U_L(t) U_F(t) \\
&= \frac{1}{d} \sum_{m=0}^{d-1} \sum_{l=0}^{d-1} \exp\left[\frac{2\pi i}{d}(pm - ql)\right] |\psi_m\rangle\langle\psi_l| \int_0^{t_0} dt \exp\{i[d(p - q) + (m - l)]\omega_0 t\} \\
&= \frac{t_0}{d} \sum_{m=0}^{d-1} \sum_{l=0}^{d-1} \exp\left[\frac{2\pi i}{d}(pm - ql)\right] |\psi_m\rangle\langle\psi_l| \delta_{pq} \delta_{ml} \\
&= \frac{t_0}{d} \sum_{m=0}^{d-1} \hat{P}_{\psi_m} \delta_{pq} \\
&= \frac{t_0}{d} \delta_{pq} \mathbf{I}
\end{aligned} \tag{2.16}$$

Portanto,

$$\begin{aligned}
\langle U_c^\dagger(t) H_{int} U_c(t) \rangle_{t_0} &= \sum_{r=0}^{d-1} \sum_{s=0}^{d-1} c_{rs} M_{rs} = \frac{t_0}{d} \sum_{r=0}^{d-1} \sum_{s=0}^{d-1} c_{rs} \delta_{rs} \mathbf{I} \\
&= \left(\frac{t_0}{d} \sum_r c_{rr} \right) \mathbf{I}
\end{aligned} \tag{2.17}$$

Portanto, nesse referencial, a hamiltoniana de interação se reduz a uma constante, que pode ser descartada.

2.2.3 GCDD no nutrit

No caso do sistema estudado neste projeto, temos $d = 3$. Neste caso, podemos chegar em H_c a partir de H_L e H_F (eq. 2.14). Desenvolvendo H_L e H_F conforme as equações 2.11 e 2.12, e eliminando o termo de $H_c(t)$ proporcional à identidade, temos:

$$\begin{aligned}
H_c(t) &= H_L + U_L(t) H_F U_L^\dagger(t) - \hbar\omega_0 I \\
&= \hbar\omega_d (|1\rangle\langle 1| + 2|2\rangle\langle 2|) \\
&+ \frac{\hbar\omega_0}{\sqrt{3}} \left[-\cos\left(\omega_d t - \frac{\pi}{6}\right) (|0\rangle\langle 1| + |1\rangle\langle 0|) + \sin\left(\omega_d t - \frac{\pi}{6}\right) (-i|0\rangle\langle 1| + i|1\rangle\langle 0|) \right. \\
&\quad - \cos\left(\omega_d t - \frac{\pi}{6}\right) (|1\rangle\langle 2| + |2\rangle\langle 1|) + \sin\left(\omega_d t - \frac{\pi}{6}\right) (-i|1\rangle\langle 2| + i|2\rangle\langle 1|) \\
&\quad \left. - \cos\left(2\omega_d t + \frac{\pi}{6}\right) (|0\rangle\langle 2| + |2\rangle\langle 0|) + \sin\left(2\omega_d t + \frac{\pi}{6}\right) (-i|0\rangle\langle 2| + i|2\rangle\langle 0|) \right]
\end{aligned} \tag{2.18}$$

Esse é o pulso de controle que iremos aplicar no nosso sistema para mitigar ruído. Mostraremos adiante, através de simulações, que conforme aumentamos o valor de n (lembrando que $t_0 = \tau/n$), a evolução do sistema de fato se aproxima à identidade.

2.2.4 Desacoplamento dinâmico no subespaço excitado

Além do GCDD, que mitiga completamente todo ruído unitário no qutrit, conforme a equação 2.18, vamos também formular um pulso que possa remover especificamente a evolução em λ_6 ou λ_7 , mantendo as outras evoluções. Vamos apenas operar dentro do subespaço excitado, isto é, $\{|1\rangle, |2\rangle\}$ de tal forma que λ_6, λ_7 sejam equivalentes a σ_x, σ_y . Vamos demonstrar que aplicando um pulso constante $H_c = n_x \hbar \omega_0 \sigma_x$ mitiga a evolução em σ_y . Começamos com uma hamiltoniana total

$$H = \hbar c_x \sigma_x + \hbar c_y \sigma_y + n_x \hbar \omega_0 \sigma_x \quad \left(= \hbar c_x \lambda_6 + \hbar c_y \lambda_7 + n_x \hbar \omega_0 \lambda_6 \right), \quad (2.19)$$

onde $\omega_0 \equiv \frac{2\pi}{t_0} = \frac{2\pi N}{\tau}$, $N \in \mathbb{N}$ é uma frequência periódica no tempo total do experimento τ . Pela equação de Schrödinger temos $i\hbar \dot{U}(t) = HU(t)$ Essa equação possui solução exata, que é

$$U(t) = \cos(\Omega t) \mathbf{I} - i \sin(\Omega t) \frac{(n_x \omega_0 + c_x) \sigma_x + c_y \sigma_y}{\Omega}, \quad \Omega \equiv \sqrt{(n_x \omega_0 + c_x)^2 + c_y^2}. \quad (2.20)$$

Tirando o limite $n_x \rightarrow \infty$ e fazendo $t = \tau$ (ou qualquer múltiplo inteiro de t_0),

$$\lim_{n_x \rightarrow \infty} U(\tau) \approx \cos[(n_x \omega_0 + c_x) \tau] \mathbf{I} - i \sin[(n_x \omega_0 + c_x) \tau] \sigma_x \quad (2.21)$$

$$\begin{aligned} &= \cos(2\pi N + c_x \tau) - i \sin(2\pi N + c_x \tau) \sigma_x \\ &= \cos(c_x \tau) - i \sin(c_x \tau) \sigma_x. \end{aligned} \quad (2.22)$$

Portanto, a dinâmica em todos os instantes de tempo segue apenas os fatores multiplicando σ_x e, em específico, a dinâmica no tempo $t = \tau$ (ou qualquer múltiplo inteiro de t_0) depende apenas do valor de c_x . O mesmo pode ser demonstrado analogamente para um DD em σ_y , isolando c_y .

Dessa forma, inserindo um pulso de controle (estático) da forma

$$H_c^{x/y} = n_{x/y} \hbar \omega_0 \sigma_{x/y}, \quad (2.23)$$

e realizando medidas em $t = \tau$, é possível isolarmos as duas frequências de ruído do *leakage*. Em específico, definindo as fidelidades:

$$F_1^{x/y} = \left| \langle 1 | U^{x/y}(\tau) | 1 \rangle \right|^2 \quad (2.24)$$

$$F_2^{x/y} = \left| \langle 1 | U^{x/y}(\tau) | 2 \rangle \right|^2 \quad (2.25)$$

$$F_3^x = \left| \langle 1 | U^x(\tau) \frac{|1\rangle - i|2\rangle}{\sqrt{2}} \right|^2 \quad (2.26)$$

$$F_3^y = \left| \langle 1 | U^y(\tau) \frac{|1\rangle + |2\rangle}{\sqrt{2}} \right|^2, \quad (2.27)$$

é trivial demonstrar que

$$F_1^{x/y} - F_2^{x/y} = \cos(2c_{x/y}t) \quad (2.28)$$

$$1 - 2F_3^{x/y} = \sin(2c_{x/y}t), \quad (2.29)$$

sendo essas equações válidas sempre que t for um múltiplo inteiro de t_0 , e tendo $n_{x/y}$ suficientemente grandes. Temos, então, um protocolo para isolar os coeficientes c_x, c_y na evolução do sistema.

Note que as eqs. 2.28, 2.29 sugerem que podemos construir uma variável complexa que contém as informações de todas as medições:

$$s^{x/y}(t) \equiv s(t) = (F_1^{x/y} - F_2^{x/y}) + i(2F_3^{x/y} - 1) \approx e^{i(2c_{x/y}t + \phi)}, \quad (2.30)$$

onde ϕ é uma fase qualquer. Assim (assumindo que ϕ se conserva ao longo do tempo, o que foi observado nas simulações), é possível encontrar os coeficientes $c_{x/y}$ realizando a medição das fidelidades em dois ou mais instantes de tempo e ajustando a reta

$$\arg(s^{x/y}(t)) = 2c_{x/y}t + \phi \quad (2.31)$$

De fato, foi utilizada a eq. 2.31 como base de desempenho para comparar com a rede neural.

2.3 Coleta de dados

Para realizar a coleta de dados, a primeira etapa foi verificar qual o valor de $n_{x/y}$ ótimo para que a equação 2.22 fosse verdadeira. De acordo com as unidades de medida estabelecidas na seção 2.1.3, o valor de ω_0 foi fixado em 2π . A fidelidade foi estabelecido de acordo com a equação de fidelidade generalizada

$$f(U_{alvo}, U) = \frac{1}{d} \left| \text{Tr}(U_{alvo}^\dagger U) \right|^2, \quad (2.32)$$

sendo $d = 3$ no caso desse sistema. Uma vez encontrado $n_{x/y}$, estabelecemos os pulsos que seriam enviados aos sistemas de acordo com a eq. 2.23. Na arquitetura do *Qiskit Dynamics*, é necessário primeiro estabelecer os canais respectivos que receberão os sinais dos pulsos (nesse caso, $\hbar\omega_0\lambda_6$ e $\hbar\omega_0\lambda_7$) e depois enviar os sinais em si, no formato de **Signals**, que neste caso são sinais estáticos ao longo do tempo.

A partir disso, foram geradas as frequências $\omega_i (i = 1, \dots, 8)$ a partir de uma distribuição uniforme variando entre 0 e $2\pi(1 \times 10^6 \text{ rad s}^{-1})$. Para cada amostra de frequências, foi gerado também um valor de α , em distribuição uniforme entre 0 e 1. A hamiltoniana do sistema foi estabelecida de acordo com a eq. 2.6.

Para cada um dos sistemas, foi efetuada a simulação da evolução com os pulsos de DD estático em λ_6 e λ_7 , conforme a eq. 2.23, com o valor de n fixado, e foram coletados dados das fidelidades F_1, F_2, F_3 , nos instantes de tempo $\tau - 0.1(1 \times 10^{-6}\text{s})$ e τ . Como foi utilizado $\tau = 4(1 \times 10^{-6}\text{s})$, então as medidas foram realizadas nos instantes $t = 3.9$ e $t = 4.0$, nas unidades da simulação.

A razão para realizar as medidas em dois instantes é remover a ambiguidade que as equações 2.28 e 2.29 têm sobre o ângulo de $s(t)$. Foi escolhido um $\delta_t = 0.1$ por duas razões. O $n_{x/y}$ ótimo encontrado foi de $n = 20$. Uma vez que $\omega_0 = 2\pi$ com período igual a 1 nas unidades da simulação, então o período da frequência $n\omega_0$ será de $\frac{1}{20} = 0.05$. Portanto, medidas válidas podem ser realizadas apenas em instantes de tempo múltiplos de 0.05. Além disso, é necessário que δ_t seja suficientemente pequeno para evitar problemas de *aliasing* devido à periodicidade de $s(t)$. Por isso foi escolhido $t = 0.1$.

Uma vez realizadas as medidas para cada amostra de frequências, estes dados foram salvos em um arquivo, com cada coluna correspondendo às simulações de uma amostra $\{\omega_i\}$. No total, foram 12 *features* medidos para cada sistema. Como o objetivo do projeto é inferir apenas o par ω_6, ω_7 , somente estes foram salvos, como objetivos (*targets*) para a rede neural e para comparação de acurácia. Além disso, foi salvo o valor de α sorteado para cada sistema, para fins de comparação.

Foram gerados 50 mil dados de treinamento mais 5 mil dados de teste utilizando o protocolo acima.

2.4 Análise com rede neural

A rede neural foi desenhada com o propósito de conseguir inferir, como alvo, o par de frequências (ω_6, ω_7) , a partir das fidelidades descritas na seção 2.2.4, utilizadas como *features*.

Foram utilizados as bibliotecas **Keras** e **TensorFlow** para o design da rede, e o ambiente do *Google Colab* para treinamento da mesma. A arquitetura geral da rede é de um MLP com quatro camadas densas para processamento com, respectivamente, 512, 256, 128 e 64 neurônios de saída cada. A primeira camada recebe como entrada os dados tratados e normalizados (o tratamento será descrito adiante), e a última camada tem como saída apenas dois neurônios, que representam o par de frequências (ω_6, ω_7) , em sua forma normalizada. As camadas são inicializadas na distribuição **He Uniform**, e têm como função de ativação o ReLu. Além disso, cada uma das camadas possui a regularização embutida L^2 com taxa de regularização $\lambda = 10^{-4}$.

Entre cada uma das camadas principais de processamento, são inseridas camadas de normalização (**BatchNormalization**) para garantir normalização entre as camadas durante o processo de treinamento, além de uma camada de *Dropout*, com taxa de 0.2, com o intuito de evitar, ao máximo possível, o *overfitting*, que é de alta relevância dada a quantidade relativamente

elevada de neurônios e conexões nessa rede.

Para o treinamento em si, a função perda utilizada foi a padrão de erro médio quadrático, com o algoritmo de propagação *Adaptive Moment Estimation* (Adam). 20% dos dados de treinamento foram reservados para validação. Durante o treinamento, para reduzir platôs e estagnação, introduzimos as funções de *callback*, ambas verificando o desempenho da função perda nos dados de validação: **EarlyStopping**, com paciência de 30, que para o treinamento caso não haja melhoria na função perda (de validação) em 30 épocas; e a **ReduceLROnPlateau**, com paciência de 8, que reduz pela metade a taxa de aprendizado cada vez que haja um platô durante 8 ou mais épocas. A taxa de aprendizado foi iniciada em 10^{-3} e foi reduzida até chegar no valor mínimo de 10^{-6} . O histórico de aprendizado da rede pode ser verificado na fig. 4.

Todo o treinamento foi realizado no espaço de *features* e alvos normalizados. Para realizar a conversão para os *features* não-normalizados, foram adicionadas camadas de normalização e desnormalização no início e final da rede, respectivamente, de tal forma que a mesma possa ser usada em qualquer ambiente, mesmo que não haja acesso aos dados de treinamento (para obter a normalização inicial).

2.4.1 Métricas de desempenho

A função perda utilizada foi a perda de Huber (L1 suave), com $\delta = 1$. Para as métricas de desempenho da rede, foi utilizada como métrica de erro a raiz do erro médio quadrático (RMSE):

$$\text{RMSE}(y, \hat{y}) = \sqrt{\text{MSE}(y, \hat{y})} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (2.33)$$

e para a métrica de acurácia, o coeficiente de determinação R^2 :

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} = 1 - \frac{\text{MSE}(\vec{Y}, \hat{Y})}{\text{Var}(\vec{Y})}, \quad (2.34)$$

onde y_i são os componentes do vetor alvo \vec{Y} e \hat{y}_i são os componentes do vetor de saída: $\hat{y}_i = F(x_i)$, onde x_i são os componentes do vetor dos *features* \vec{X} e F é a rede neural.

Escolhemos R^2 como métrica de "acurácia" da rede. Para o modelo perfeito, $R^2 = 1$, que significa que toda variação dos dados é explicada pelo modelo; se o modelo for trivial (prever apenas a média dos dados), teríamos $R^2 = 0$, e se ele for pior do que isso, $R^2 < 0$.

2.5 Síntese

Estabelecemos, portanto, um protocolo para mitigação de ruído *leakage* em um qubit *transmon*. O protocolo consiste em: aplicação de pulsos descritos na seção 2.3 e inferência das frequências de *leakage* com uso da rede neural. A partir disso, estabelecemos a frequência

máxima para a qual será aplicado o DD: $\omega_{max} = \max(\omega_6, \omega_7)$, fazendo $\omega_0 = \omega_{max}$, determinando assim os pulsos de DD que serão enviados ao qubit, de acordo com a eq. 2.18.

3 RESULTADOS

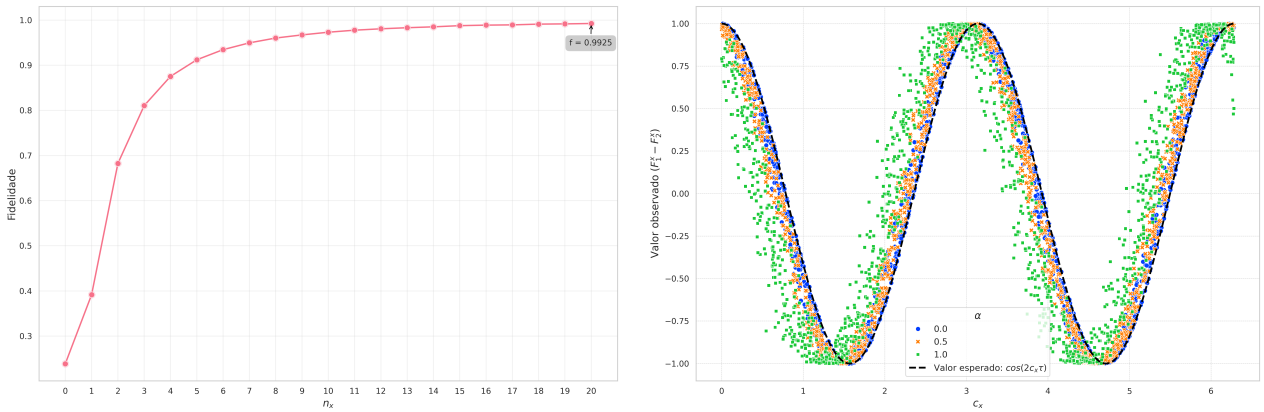
3.1 Verificação do desacoplamento dinâmico estático

Primeiramente, após a configuração do ambiente de simulação, foi realizada a verificação do processo descrito na seção 2.2.4, utilizando como fidelidade função descrita na eq. 2.32. A curva da fidelidade em função do valor de n_x , para o desacoplamento dinâmico em σ_x , pode ser visualizada na figura 3a. A curva para σ_y e n_y é idêntica.

Com o intuito de verificar a equação 2.28 para diferentes valores de α , foram gerados 1500 amostras de frequências $\{\omega_i\}$, para $\alpha = 0$ (sem frequências não-*leakage*), $\alpha = 0.5$ (outras frequências com metade da intensidade) e $\alpha = 1$ (mesma intensidade). Para cada um desses sistemas, foram medidas as fidelidades F_1^x e F_2^x . Os resultados podem ser verificados na figura 3b. Foi observado que, conforme α cresce, a distribuição ainda mantém o formato de cosseno, porém adquire uma fase, positiva, que cresce em conjunto com α . Além disso, a distribuição fica mais dispersa, variando mais em relação à sua média.

3.2 Treinamento e desempenho da rede neural

A rede neural foi treinada a partir do conjunto de 50000 dados das fidelidades, de acordo com o protocolo descrito na seção 2.3. O histórico de treinamento da rede pode ser visualizado na fig. 4. Foram gerados 5000 dados de para fazer parte do conjunto de teste (dados com os quais a rede nunca teve contato).



- (a) Fidelidade do DD estático em σ_x , com destaque para a fidelidade em $n_x = 20$. Nessa simulação, $c_x = c_y = 2\pi$. (b) Verificação da eq. 2.28 para o DD em σ_x , para diferentes valores de α , a ordem de grandeza relativa das frequências não-*leakage*. Aumentar α adiciona uma fase ao cosseno, além de adicionar incerteza.

Figura 3 – Simulação e verificação das equações 2.22 e 2.28.

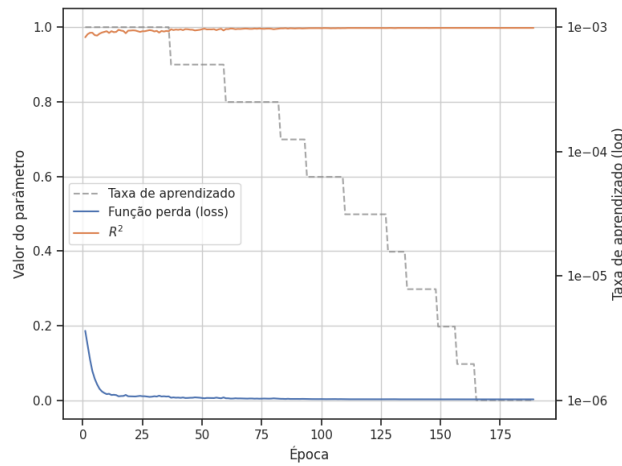


Figura 4 – Histórico de treinamento da rede neural.

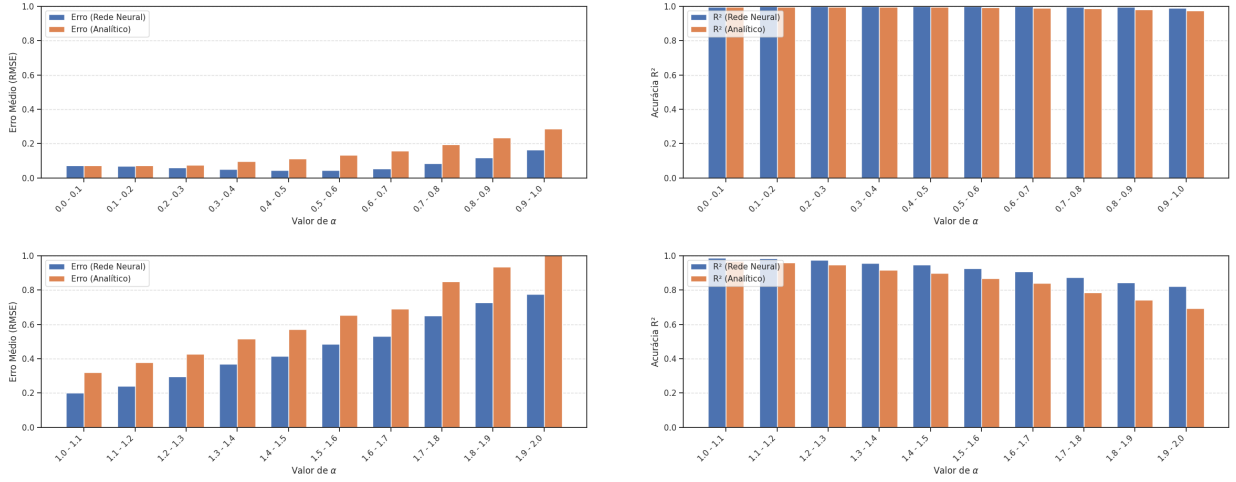
Além disso, foi verificado o desempenho do método analítico (vide eq. 2.31). O desempenho médio da rede no conjunto de teste foi muito bom, com erro médio $RMSE = 0.08$ e acurácia $R^2 = 0.998$. Em comparação, o método analítico obteve erro médio de $RMSE = 0.16$ e acurácia $R^2 = 0.992$. Ambos tiveram acurácia muito boa.

Agrupando o conjunto de teste com base nos valores de α , foi observado que a rede neural teve desempenho consistentemente melhor do que o método analítico para $\alpha > 0.5$, e a disparidade entre o desempenho dos dois cresceu conforme α aumentou. Para verificar essa relação, foram gerados também dados com $\alpha \in (1, 2)$, sistemas com aos quais a rede neural não havia sido exposta no seu treinamento. Mesmo assim, a rede apresentou desempenho consideravelmente melhor do que o método numérico, indicando a adaptabilidade do método de redes neurais para resolver problemas desta espécie. A comparação entre os dois métodos pode ser visualizada na fig. 5.

3.3 Mitigação de ruído com GCDD

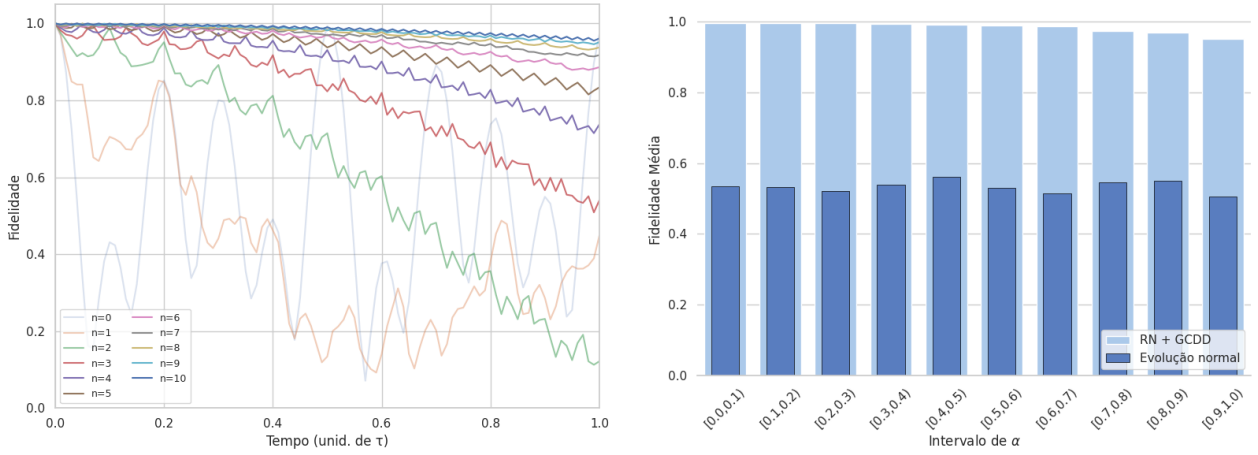
Foram realizadas simulações para verificar o protocolo de desacoplamento dinâmico contínuo, descrito nas seções 2.2.2 e 2.2.3. O desempenho dos pulsos na eq. 2.18, para diferentes valores de n , foi verificado (para frequências de ruído $\{\omega_i\}$ escolhidas arbitrariamente) e pode ser visualizado na figura 6a. A partir dessa análise, estabelecemos $n = 10$ como equilíbrio entre energia do pulso (diretamente proporcional a n) e fidelidade satisfatória.

Depois, foi verificado o desempenho do protocolo RN + GCDD (descrito na seção 2.5), para mitigar o ruído em um sistema arbitrário. O desempenho desse protocolo foi extremamente satisfatório, apresentando alta mitigação de ruído, com fidelidade média acima de 0.95 para $\tau = 5\mu s$ e todos os valores de α , sendo a fidelidade média acima de 0.99 para $\alpha \approx 0$. Os resultados podem ser visualizados na figura 6b.



- (a) Comparação do erro médio entre a rede neural e o método analítico. (b) Comparação do coeficiente de determinação R^2 entre a rede neural e o método analítico.

Figura 5 – Comparação entre a rede neural e o método analítico.



- (a) Desempenho de fidelidade do GCDD para diferentes valores de n . A fidelidade final aumenta consideravelmente com cada acréscimo em n . Com $n = 10$, a fidelidade final é cerca de 0.95. Tempo total de simulação $\tau = 5\mu s$. (b) Fidelidade média para diferentes valores de α , com tempo de ociosidade $\tau = 5\mu s$ e aplicação de pulsos GCDD, sendo $\omega_0 = \omega_{max}$ a frequência máxima de ruído prevista pela rede neural. O GCDD apresenta mitigação quase completa do ruído, com fidelidade muito acima da sem mitigação de ruído.

Figura 6 – Desempenho do GCDD.

4 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Concluimos que o protocolo proposto neste projeto, de espectroscopia de ruído via medições específicas de fidelidade, inferência dos parâmetros do sistema por meio de uma rede neural, e mitigação de todo ruído unitário por meio do desacoplamento dinâmico contínuo, é tecnicamente sólido e efetivo no regime simulado de *leakage* estático. Modelando o qubit *transmon* com três níveis de energia e parametrizando o ruído por meio das matrizes de Gell-Mann, a rede (MLP) alcançou acurácia altíssima ($R^2 > 0,998$) na predição das frequências (ω_6, ω_7) e manteve desempenho elevado mesmo na presença de perturbações adicionais de ordem $\alpha = 1$. Comparação com um método analítico demonstrou superioridade da rede neural em quase todas as situações.

Integrado a isso, o controle com desacoplamento dinâmico contínuo mostrou fidelidades próximas da unidade em toda a evolução para $n \gtrsim 10$ (com $\tau = 5 \mu s$), corroborando que a estimativa correta da banda de ruído e, em particular, da maior frequência, é suficiente para definir ω_0 e projetar pulsos que suprimem, na prática, erros unitários relevantes. Além de ser compatível com o ecossistema atual do *Qiskit*, o fluxo é reproduzível e modular, permitindo a troca de modelos e a extensão para outros observáveis de fidelidade. Os resultados indicam que poucas medidas bem escolhidas, seguidas de inferência orientada por estrutura física e controle contínuo, bastam para suprimir, na prática, erros unitários de *leakage* em regimes NISQ.

Como próximos passos, propomos validar o protocolo em plataformas com acesso a controle de pulsos e estender a identificação para ruído lentamente dependente do tempo e para sistemas de dois qubits. Dado o estado de manutenção do pacote *Qiskit Dynamics*, recomendamos fortemente que projetos futuros na área portem a implementação para ecossistemas mais ativos.

REFERÊNCIAS

- 1 FEYNMAN, R. P. Simulating physics with computers. **International Journal of Theoretical Physics**, v. 21, n. 6–7, p. 467–488, 1982. Disponível em: <https://vql.cs.msu.ru/Feynman.pdf>.
- 2 GROVER, L. K. A fast quantum mechanical algorithm for database search. In: **Proceedings of STOC '96**. [S.l.: s.n.], 1996. p. 212–219. Disponível em: <https://dl.acm.org/doi/10.1145/237814.237866>.
- 3 SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM Journal on Computing**, v. 26, n. 5, p. 1484–1509, 1997. Disponível em: <https://epubs.siam.org/doi/10.1137/S0097539795293172>.
- 4 CAO, Y.; ROMERO, J. *et al.* Quantum chemistry in the age of quantum computing. **Chemical Reviews**, v. 119, n. 19, p. 10856–10915, 2019. Disponível em: <https://pubs.acs.org/doi/10.1021/acs.chemrev.8b00803>.
- 5 MCARDLE, S. *et al.* Quantum computational chemistry. **Reviews of Modern Physics**, v. 92, n. 1, p. 015003, 2020. Disponível em: <https://link.aps.org/doi/10.1103/RevModPhys.92.015003>.
- 6 CHOW, J. C. L. Quantum computing in medicine. **Medicina**, v. 12, n. 4, p. 67, 2024. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11586987/>.
- 7 EISERT, J.; PRESKILL, J. **Mind the gaps: The fraught road to quantum advantage**. 2025. Os autores afirmam: "*quantum computations that are practically useful and economically viable have not yet been achieved, nor is it clear when that might happen.*". Disponível em: <https://arxiv.org/abs/2510.19928>.
- 8 PRESKILL, J. Quantum computing in the nisq era and beyond. **Quantum**, v. 2, p. 79, 2018. Disponível em: <https://quantum-journal.org/papers/q-2018-08-06-79/>.
- 9 IBM Quantum. **The era of quantum utility is here**. 2023. Blog/Página institucional. Anuncia o processador *IBM Condor* de 1 121 qubits. Disponível em: <https://www.ibm.com/quantum/blog/quantum-roadmap-2033>.
- 10 IBM Quantum. **Technology and roadmap**. 2025. Página institucional. Disponível em: <https://www.ibm.com/quantum/technology>.
- 11 Quantinuum. **Quantinuum's H-Series hits 56 physical qubits that are all-to-all connected**. 2024. Blog. Disponível em: <https://www.quantinuum.com/blog/quantinuums-h-series-hits-56-physical-qubits-that-are-all-to-all-connected-and-departs-the-era-of-classical-simulation>.
- 12 AASEN, D. *et al.* **Roadmap to fault tolerant quantum computation using topological qubit arrays**. 2025. Disponível em: <https://www.microsoft.com/en-us/research/publication/roadmap-to-fault-tolerant-quantum-computation-using-topological-qubit-arrays/>.
- 13 SETHARES, W. A.; COAUTHORS. Instead of 'supremacy' use 'quantum advantage'. **Nature**, v. 576, p. 213, 2019. Disponível em: <https://www.nature.com/articles/d41586-019-03781-0>.
- 14 The Nobel Prize. **Press release: The Nobel Prize in Physics 2025**. 2025. Press release. Prêmio a John Clarke, Michel H. Devoret e John M. Martinis; contexto de circuitos supercondutores. Disponível em: <https://www.nobelprize.org/prizes/physics/2025/press-release/>.

- 15 The Royal Swedish Academy of Sciences. **Scientific Background to the Nobel Prize in Physics 2025**. 2025. Documento técnico. Discute cQED e o *transmon* como design insensível a ruído de carga. Disponível em: <https://www.nobelprize.org/uploads/2025/10/advanced-physicsprize2025.pdf>.
- 16 KRINNER, S. *et al.* Engineering cryogenic setups for 100-qubit scale superconducting circuit systems. **EPJ Quantum Technology**, v. 6, n. 1, p. 2, 2019. Exemplo de operação a ~ 14 mK em setups supercondutores. Disponível em: <https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-019-0072-0>.
- 17 DUTTA, A.; SARMA, N.; KARMAKAR, A. A survey of noise in quantum computing. *In: 2025 IEEE Guwahati Subsection Conference (GCON)*. [S.l.: s.n.], 2025. p. 1–6.
- 18 CHATTERJEE, A. **Quantum Error Correction For Dummies**. 2023. Disponível em: <https://arxiv.org/abs/2304.08678>.
- 19 RAHMAN, A.; EGGER, D. J.; ARENZ, C. **Learning How to Dynamically Decouple**. 2024. Disponível em: <https://arxiv.org/abs/2405.08689>.
- 20 HYYPPÄ, E. *et al.* **Reducing leakage of single-qubit gates for superconducting quantum processors using analytical control pulse envelopes**. 2025. Disponível em: <https://arxiv.org/abs/2402.17757>.
- 21 NAPOLITANO, R. d. J. *et al.* Protecting operations on qudits from noise by continuous dynamical decoupling. **Physical Review Research**, v. 3, n. 1, 2021. ISSN 2643-1564. Disponível em: <http://dx.doi.org/10.1103/PhysRevResearch.3.013235>.
- 22 MOTTA, O. d. **Aspectos Teóricos de Aplicações do Desacoplamento Dinâmico Contínuo**. 2024. Dissertação (Tese de mestrado) — Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2024. Disponível em: <https://doi.org/10.11606/D.76.2024.tde-20092024-093246>.
- 23 STRAUCH, F. W. **Dephasing-induced leakage in multi-level superconducting quantum circuits**. 2025. Disponível em: <https://arxiv.org/abs/2501.17008>.
- 24 NIELSEN, M. A. Neural networks and deep learning. *In: _____*. [S.l.: s.n.]: Determination Press, 2015. cap. 4.
- 25 JAVADI-ABHARI, A. *et al.* **Quantum computing with Qiskit**. 2024. Disponível em: <https://arxiv.org/abs/2405.08810>.
- 26 PUZZUOLI, D. *et al.* Qiskit dynamics: A python package for simulating the time dynamics of quantum systems. **Journal of Open Source Software**, The Open Journal, v. 8, n. 90, p. 5853, 2023. Disponível em: <https://doi.org/10.21105/joss.05853>.
- 27 IBM. **Migrate from Qiskit Pulse to fractional gates**. 2025. Disponível em: <https://quantum.cloud.ibm.com/docs/en/migration-guides/pulse-migration>.
- 28 LAMBERT, N. *et al.* Qutip 5: The quantum toolbox in python. 2024. Disponível em: <https://arxiv.org/abs/2412.04705>.
- 29 AMAZON. **Pulse control on Amazon Braket**. 2025. Disponível em: <https://docs.aws.amazon.com/braket/latest/developerguide/braket-pulse-control.html>.
- 30 LANGE, F. *et al.* **Cross-talk in superconducting qubit lattices with tunable couplers - comparing transmon and fluxonium architectures**. 2025. Disponível em: <https://arxiv.org/abs/2504.10298>.

31 CHIARO, B.; ZHANG, Y. **Active leakage cancellation in single qubit gates**. 2025. Disponível em: <https://arxiv.org/abs/2503.14731>.

32 GOIPORIA, P. *et al.* **Suppressing errors with dynamical decoupling using pulse control on Amazon Braket**. 2022. Disponível em: <https://aws.amazon.com/blogs/quantum-computing/suppressing-errors-with-dynamical-decoupling-using-pulse-control-on-amazon-braket/>.