Effects of T1 Noise on the Deutsch–Jozsa Algorithm

Stephen Aranda
University of Texas at Dallas
PHYS 4346 – Quantum Mechanics for Engineers and Programmers

Abstract

In this project, we will examine the effects of noise on the Deutsch-Jozsa algorithm. The purpose of this project is to use the Deutsch-Jozsa algorithm as a proxy to examine how an algorithm and a quantum computer are affected by noise. From this, the hope is that the insights gained from examining and visualizing the effects of the noise will lead to a potential solution or mitigation of this issue. For the purposes of time constraints, the noise that will be examined in the experiment will be T1 noise. Five trials will be performed in which T1 noise will be injected in a two-qubit rho density matrix that represents an initial ground state after the final quantum gate in the algorithm is applied. Afterwards, the results of the trials will be discussed and then a suggestion will be made.

1 Introduction

In 1992, David Deutsch and Richard Jozsa, a physicist from the University of Oxford and a mathematician from the University of Cambridge respectively, proposed a quantum algorithm called the Deutsch-Jozsa Algorithm. This algorithm was one of the first algorithms created for a quantum computer for the sole purpose of demonstrating how a quantum algorithm, and a quantum computer, could be used to solve a problem much faster than a traditional algorithm being executed on a classical, deterministic computer.

Despite the advantages that quantum computers offer in terms of speed increases in computation tasks, there are numerous issues that persist in preventing quantum computers from being fully realized in our everyday lives like classical computers. Examining one of these issues to gain greater insight into how to potentially solve these problems is the main motivator behind this research project.

2 Overview of the Deutsch–Jozsa Algorithm

For the purpose of informing the reader who is not familiar with the Deutsch-Jozsa algorithm, a brief overview of this algorithm will be given. The problem that the algorithm is meant to solve is to solve a simple problem of figuring out what type of function is contained by a black box known as the oracle. For those unfamiliar with the term “black box”, it simply means you have some type of computer system that the internal structure is not known except for the input and output. The function that is contained in the oracle is a Boolean function that is either a constant function or a balanced function. If the function is constant, an evaluation will be performed on the inputs received by the oracle in the form of input qubits and the result is the state of the input qubits is not affected after the evaluation completes execution. Hence, why this is regarded as a constant function. Because the state of the input qubits remains constant. The output value of the function is 0. This value will be held by an output qubit to inform that this is a constant function. If the function is a balanced function, then a flip operation occurs in which half of the input qubits are flipped to 1 and the other half are 0. The result of the function is 1, which is held by the output qubit.

Here is how the algorithm gains the speed advantage over a classical computer. The algorithm’s circuit utilizes a Hadamard gate before entering the oracle. The Hadamard gate applies an evenly distributed superposition on all possible states on the input qubits and the output qubit. After exiting the oracle, a final Hadamard gate is applied to the input qubits and a measurement is taken on the states of the input qubits. Since all possible states are being taken into consideration at the same time, this means that there needs to be only one evaluation versus multiple evaluations on a classical machine. This is how speed increase occurs with this particular algorithm, because of the utilization of superposition. Now we will discuss the platform that was utilized for this experiment.

3 Platform for Experiment

This project was conducted via Qiskit. Qiskit is IBM’s platform called ”Quantum Lab” that is used to work with quantum computers. This platform has many pre-built functionalities that allow a quantum circuit to be built, via Python programming language, for the purpose of running this algorithm for examination. But because of the decision to conduct this project in the form of matrices and matrix operations, many of these pre-built functionalities involving quantum circuits, simulators, and pre-built noisy models were not used. Everything was realized by calculations performed by hand and then implemented in Qiskit.

4 Implementation of the Algorithm

The implementation follows the description presented in *Quantum Computation and Quantum Information*, with several modifications. A two-qubit density matrix formalism was used instead of an n-qubit pure state representation. Oracle outputs were indicated using classical print statements rather than output qubits. After the final Hadamard gate, T1 noise was injected via iterative time evolution of the density matrix.

5 Realization of Quantum Gates

Quantum gates were realized by constructing Hamiltonians and applying the unitary time evolution operator

$$U(t) = e^{-iHt/\hbar}. \tag{1}$$

The gates that were used were the Hadamard gate and Z-gate.

5.1 Hadamard Gate

The single-qubit Hadamard Hamiltonian used in this experiment is

$$H = -\frac{b}{2\sqrt{2}}(X + Z). \tag{2}$$

This Hamiltonian was tensored to construct a two-qubit operator. Using a π-rotation time $t = \pi\hbar/b$, the Hadamard gate was physically realized and applied to both qubits.

$$H_{12} = H \otimes H \tag{3}$$

5.2 Z Gate

The Z gate Hamiltonian for a single qubit is

$$H_Z = \frac{b}{2}Z. \tag{4}$$

To flip only the second qubit, this Hamiltonian was tensored with the identity operator. The resulting unitary was applied using the same π-rotation time.

$$H_{Z2} = I \otimes H_z. \tag{5}$$

6 T1 Noise Modeling

T1 noise was modeled using the Lindblad master equation,

$$\frac{d\rho}{dt} = \gamma \left(\sigma_- \rho \sigma_+ - \frac{1}{2}\{\sigma_+\sigma_-, \rho\} \right). \tag{6}$$

Modified ladder operators were constructed by tensoring to target relaxation of the second qubit only. Noise was injected after the final Hadamard gate, corresponding to the point at which an excited state is present.

7 Experimental Trials

7.1 Trial One: Ideal Scenario

An ideal, noise-free trial was performed for both constant and balanced oracle evaluations.

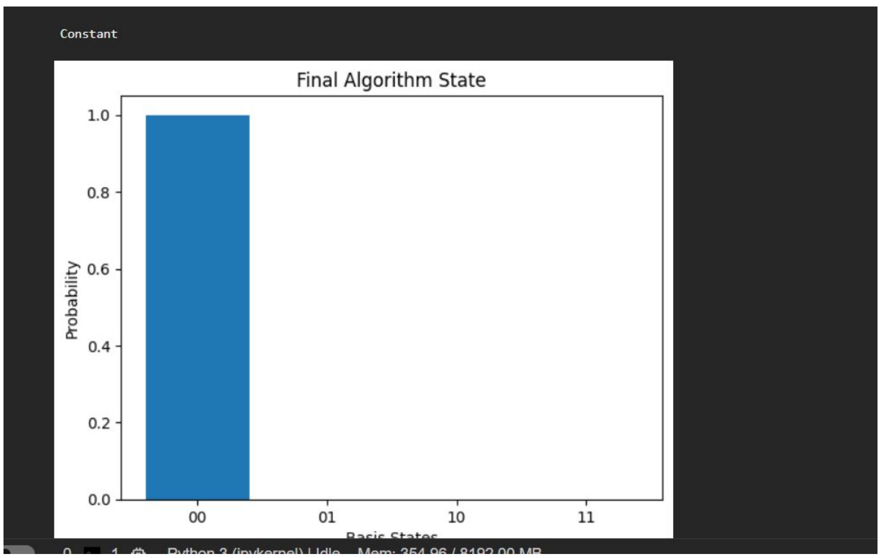

Figure 1: Ideal constant-function evaluation showing the ground state after measurement.

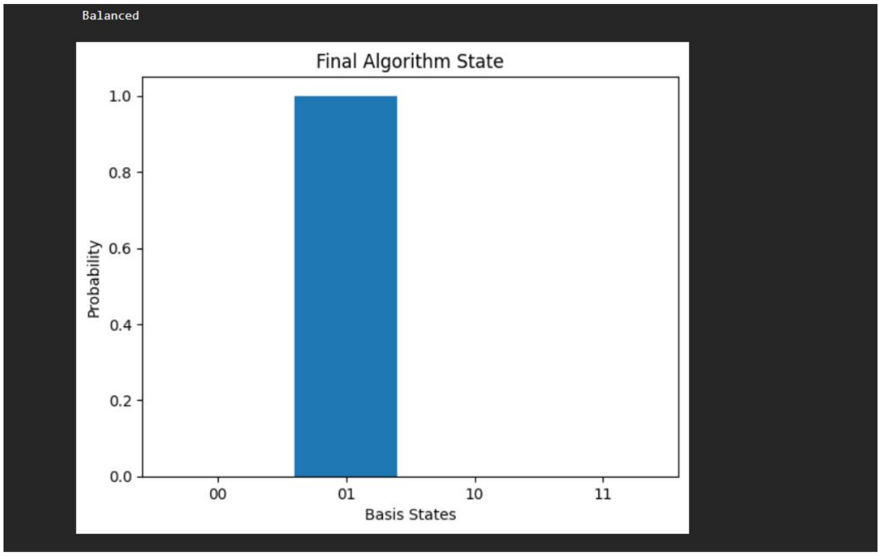

Figure 2: Ideal balanced-function evaluation showing the excited $|01\rangle$ state.

7.2 Trials Two Through Five

T1 noise was injected for increasing durations following the final Hadamard gate. Trials ranged from one second to four seconds of relaxation time.

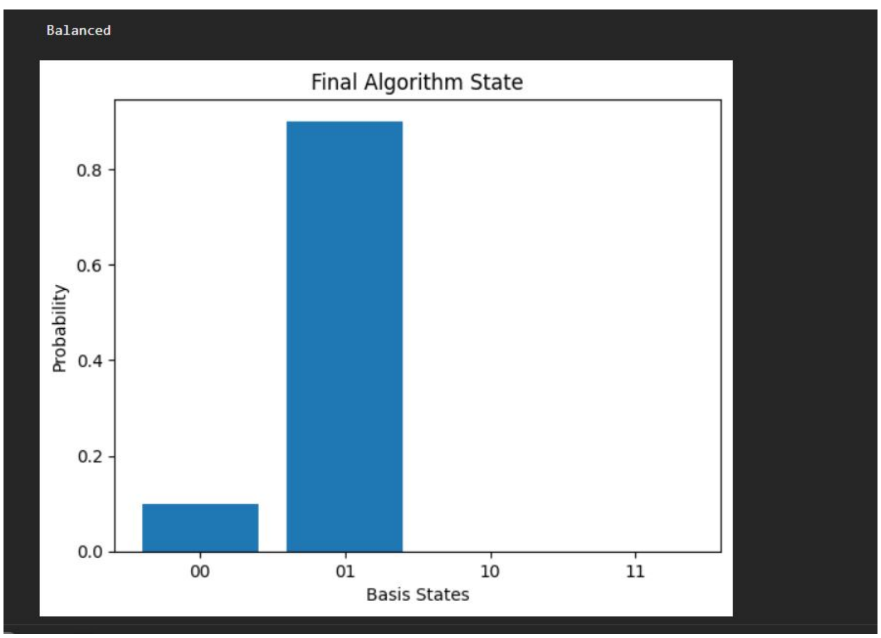

Figure 3: Balanced-function evaluation with one second of T1 noise injection.

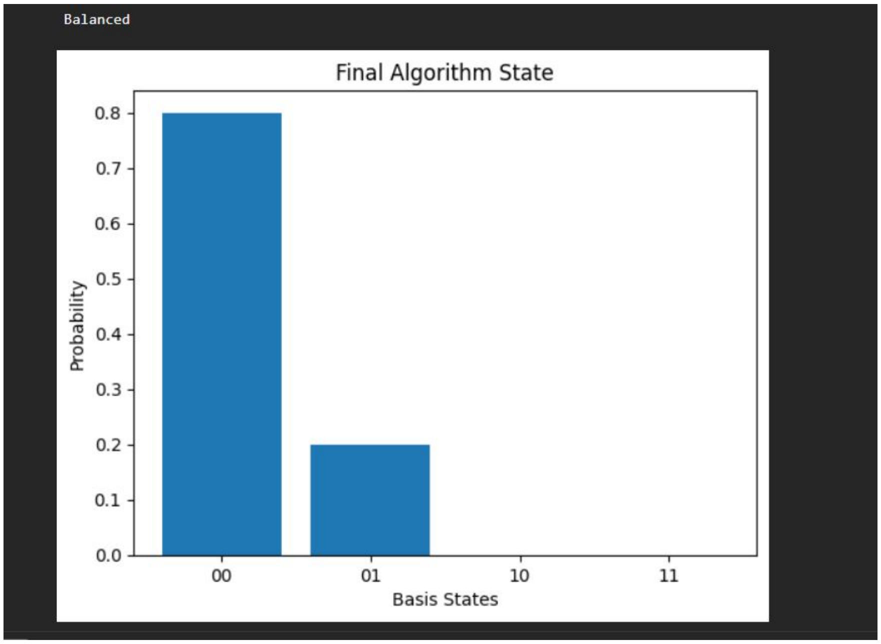

Figure 4: Balanced-function evaluation with four seconds of T1 noise injection, showing full relaxation.

8 Discussion and Mitigation

In this experiment, which one could argue is a noise simulation, T1 noise was injected into the Deutsch-Jozsa algorithm. From this experiment, we have a visualization of the relaxation of the excited qubit. The question now becomes, after having this visual and researching, is there a hardware or software solution to either eliminate or, at the very minimum, mitigate the noise to a tolerable level?

While there are multiple mitigation techniques that were found during research, one technique that was found to be promising was quantum error correcting code. This suggestion comes about as a result of reviewing the for loop that was created for the noise injection into the density matrix. During the initial implementation of this algorithm, test trials were giving incorrect results in the form of probabilities greater than 1 and below 0. It took a great deal of time to solve this problem but eventually a breakthrough was made when a realization was thought of after analyzing the results of the noise injection. The density matrix needed to be normalized. After conducting several tests and doing more research, a normalization constant was used in the form of dividing the injected density matrix by its trace. This ensured that the density matrix remained normalized through the trials involving noise injection.

Thinking about this experience drove research into error correcting code. According to Zaira Nazario (Nazario, sec Correcting Errors), helper qubits can be utilized in error correcting code by entangling them with the actual qubits that represent your quantum information. This is very relevant to the mitigation of the noise used in this experiment because Nazario continues to explain that through this entanglement, noise that may exist within your actual qubits are going to be transferred to the helper qubits. By transferring the noise to the helper qubits, you can then measure your helper qubits to get the information you need to find the errors and fix them without affecting your actual qubits and therefore not destroying your quantum information. After reviewing this article, this leads to the conclusion that quantum error correcting code is a very good candidate as a mitigation tool to deal with T1 noise.

9 Conclusion

This project analyzed the effects of T1 noise on the Deutsch–Jozsa algorithm using Hamiltonian-based gate construction and Lindblad dynamics. The results provide insight into algorithmic degradation due to relaxation and motivate the use of quantum error correction as a viable mitigation technique.

References

Z. Nazario, “How to Fix Quantum Computing Bugs,” *Scientific American*, 2022.

O. Dioti et al., “Normalization Constant in Time Evolution of Density Matrix,” Physics Stack Exchange.