COMS W3998 Final Project: Quantum-based Pattern Recognition Algorithm for Gravitational Wave Search

Manuel Paez

May 11th 2023

1 Introduction

Quantum machines present an opportunity to revolutionize computing by executing calculations faster than what is computationally possible for classical computers. One fundamental computing task that may require vast computational resources to execute is data recognition of known or unknown patterns. With the rapid amount of gathered information being processed, faster pattern recognition could dramatically expand analysis and execution capabilities. We implement a new qubit-based matched filtering algorithm on noisy superconducting qubits to carry out the first quantum-based search for gravitational waves. We obtain results that show that a quantum algorithm that obtains a signal-to-noise ratio akin to a classical algorithm that obtains a signal-to-noise ratio.

2 Project

The outline of this project was to create a coding implementation of a qubit-based matched filtering algorithm design to carry out a quantum-based search for gravitational. The equations and Algorithms were already designed thanks to Szabolc Marka and Doğa Veske. However, tests on a quantum simulator and quantum hardware needed to be generalized for repeated use. With this, the goal of this project is as follows:

1. Implement the quantum-advantage pattern recognition algorithm on a quantum machine simulator
2. Implement the quantum-advantage pattern recognition algorithm on a real quantum machine

We implemented our results using Python and the IBM Qiskit library. We also used IBM Quantum Lab to not only code-up our algorithm on a jupyter notebook but to generate results for our algorithm through its friendly user interface for submitting tasks to real IBM devices. For Part 1, we used Qiskit's implemented fake simulator "ibm_oslo", a simulator of the recently retired "ibm_oslo" machine to generate results for our implemented algorithm. Since fake simulators allow us to obtain approximate results in a short time, we do this extensively in the testing phase of our implementation. For Part 2, we implemented our results on the real device "ibmq_belem", a 5-qubit device a part of IBM Research.

For further details on our implementation of the algorithm, we created three functions, *sub_circ*, *circuit_builder* and *snr*. From the data and signal values provided, we would use *cirucit_builder* to build a collection of transpile circuits from the function *sub_circ* with the data and signal values provided to

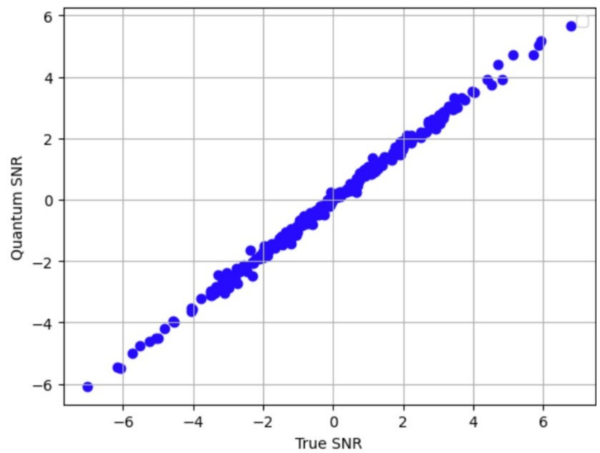

Figure 1: True SNR vs Quantum SNR
(simulator)

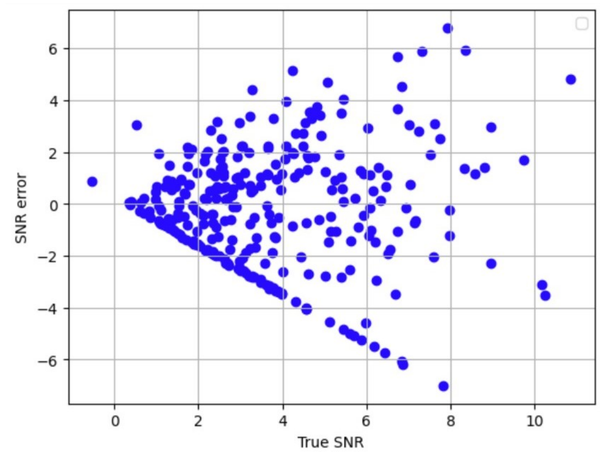

Figure 2: True SNR vs error
(simulator)

create one gigantic circuit. The function *sub_circ* takes in data and signal values such that on 4 qubits and 3 classical bits, we generate circuitry by calculating the encoding angles from the data and signal values. These tasks would be sent to the IBM machine to run. Afterwards, we output the True (classical) SNR results and the Quantum SNR results in the function *snr*. We constructed functions such that the retrieval of jobs submitted to IBM machines would be easy access to the common user. Thus, this completes the project.

3 Results

Note: Results and Implementation can be seen here: [MP23]
Using a Gaussian randomly generated number set for *data*, we were able to obtain results that correspond to calculated SNR for both real-life quantum and simulated quantum devices. We generated graphs to compare the Classical SNR results with the Quantum SNR results and the Classical SNR results with the error we need to counteract for in the Quantum Algorithm. Although many other graphs can be given, I believe these reflect the outcome the best while also serving as a comparison for the graphs on the draft paper. For Part 1, we were able to generate results on a simulated device that showed that their classical SNR and the Quantum SNR were nearly the same. We also showed the error created from the Quantum SNR compared to the error in the True SNR in relation to the Algorithm. This is done since for calculating the Quantum SNR, there will be an error that must be calculated and used to obtain the actual results of the algorithm. The results from the simulated device are shown in Figures 1 and 2. For Part 2, we were able to generate results on a real device akin to the results shown in the simulator. Since these are real computers, we notice much more errors and noise in the real-device results than with the simulation. This is partly due to the errors created from actual quantum hardware, as the simulator is mostly fine-tuned. However, the trends remain the same. The results of the real-device is shown in Figures 3 and 4.

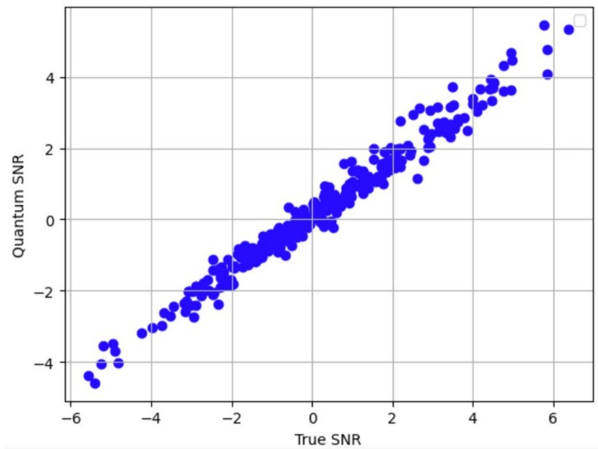

Figure 3: True SNR vs Quantum SNR
(Real Device)

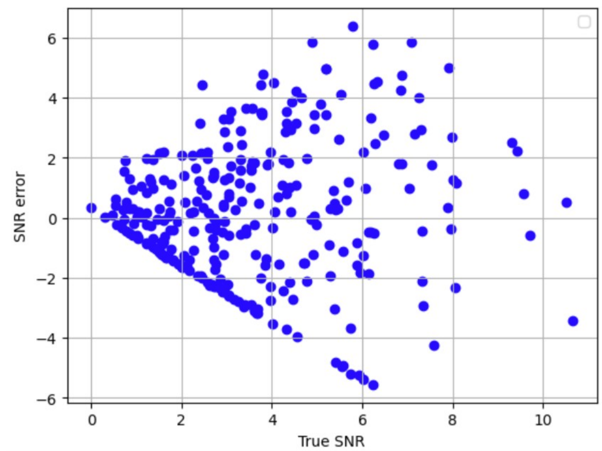

Figure 4: True SNR vs error
(Real Device)

4 Conclusion

We have shown an implementation of a new qubit-based matched filtering algorithm on noisy superconducting qubits to carry out the first quantum-based search for gravitational waves. We have also shown that this implementation on real quantum qubits is able to generate results that are compatible with results from a classical computer. Although the Algorithm only runs on 4-noisy qubits, we hope to expand on this for greater run times in the future. Thus, we have implemented a new qubit-based matched filtering algorithm on noisy superconducting qubits to carry out the first quantum-based search for gravitational waves. I want to thank Professor Szabolc Márka and Doğa Veske for mentoring me and helping me over the course of the term, and I want to thank IBM for generously lending their quantum machines for us to obtain the results provided in the paper.

References

[MP23] Department of Computer Science at Columbia University of City in New York Manuel Paez. Quantum pattern recognition algorithm for gravitational wave search, 2023.