QFT Symphony: Digital Signal Processing in the Quantum Era

Akash Malemath, Guided By: Prof. Aurelien Coillet

[a] *UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ (UBFC), Dijon,*

Abstract

Digital Signal Processing is pivotal in numerous technological applications, from day-to-day telecommunications to detecting gravitational waves. Traditionally, we leverage various classical algorithms as techniques for digital signal processing. Since few years, there has been a paradigm shift towards using quantum computation and quantum algorithms to find the quantum advantage over classical. This paper explores the application of one such quantum algorithm, Quantum Fourier Transform (QFT), which can be a substitute for current classical techniques, in the realm of Digital Signal Processing.

Keywords: Discrete Fourier Transform, Frequency Spectrum, Spectrogram

1. Introduction

Quantum Computing has a fundamental new approach towards solving computational tasks, unlike digital computers. They broadly work the same way, by taking the input, performing a finite number of elementary operations and producing the output, but the rules of computation for each of these steps are completely different. These rules given by quantum mechanics, allows to gain computational advantage in solving some problems much faster. Typically, the quantitative data can be classified into three fundamental types, *analog, digital* and *quantum data*. While both *analog* and *digital* uses real numbers, *quantum data* uses complex numbers. The compatibility of *quantum data* with other types of data is what helps in quantum computing.

Unlike classical computers, quantum computers use quantum bits called Qubits, which is similar to a probabilistic bit, i.e., a qubit can have a $|\alpha|^2$ probability to be in 0 and $1-|\alpha|^2$ probability to be in 1. Mathematically, it can be written as, $\alpha\,|0\rangle+(1-\alpha)\,|1\rangle$, where $|0\rangle$ and $|1\rangle$ are equivalent representations of classical bits 0 and 1 in Hilbert space. This superposition of having a qubit to be in multiple states at the same time is what helps us to solve a problem with fewer qubits compared to classical bits. As a single qubit can be either in $[|0\rangle, |1\rangle]$, two qubits can be in $[|00\rangle, |01\rangle, |10\rangle, |11\rangle]$, a data of size N would just require $\log_2(N)$ qubits to encode it in a quantum circuit.

The classical computation utilizes the set of logical gates to perform tasks while quantum computers have their own quantum gates for the same. Any operation can be realized as the quantum gate, iff it is a unitary (reversible) operation. Few of the well known quantum gates include Hadamard gate, Controlled-NOT gate, Rotation gates and more. Any multiqubit unitary operation can be decomposed into consecutive action, mathematically tensor product of such quantum gates. After we perform the task on a quantum circuit, we measure the qubits in order to collapse the superposition into a particular classical state to read out the output. We repeat the measurements many times by taking the copy of the quantum circuit in order to get probabilities discussed before.

2. Quantum Fourier Transform

Quantum Fourier Transform (QFT)(1) is a well known quantum algorithm which is employed in many other quantum algorithms, most notably Shor's algorithm and Quantum Phase Estimation algorithm. QFT is a quantum implementation of the discrete Fourier transform over the amplitudes of wavefunction. Thus we can use the QFT as the basis of the study of signal processing.

Fourier transform can be defined as the method to uncover hidden periodic structures in a signal. In digital signal processing, we make use of the Discrete Fourier Transform (DFT), by mapping the signal from time domain to frequency domain. Using DFT, we can effectively express every vector (i.e., every signal) in time domain, as a linear combination of simple periodic vectors (i.e., sinusoidal signals) in frequency domain. Visual representation of this can be seen in fig 1.

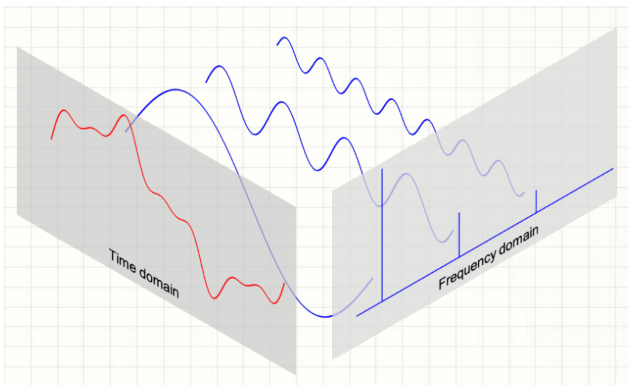

Figure 1: Fourier Transform.

Mathematically, DFT is given as

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i jk}{\sqrt{N}}}$$

where y_k is the transformed data (output vector of complex numbers) and x_j is the input vector of length N.
Now, writing the quantum equivalent of the above equation

$$|j\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{\frac{2\pi i jk}{\sqrt{N}}} |k\rangle$$

This is the origin of QFT. Let us try to visualize the equivalence between QFT and DFT, by considering an arbitrary vector (i.e., signal of length N) $|x\rangle$ which can be written as the linear combination of basis vectors $|j\rangle$, then

$$|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle = \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_j e^{\frac{2\pi i jk}{\sqrt{N}}} |k\rangle$$

$$|x\rangle = \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i jk}{\sqrt{N}}} \right) |k\rangle$$

$$|x\rangle = \sum_{k=0}^{N-1} y_k |k\rangle = |y\rangle$$

where $|y\rangle$ is the output transformed vector with fourier coefficients y_k. In QFT we take the length N = 2^n, where n is some integer so that the basis vectors $|j\rangle$ are the computational basis $|0\rangle, .., |2^n - 1\rangle$.

It might not be obvious from the above definition, but this transformation is a unitary transformation and thus can be implemented as the dynamics of a quantum state in a quantum computer. The corresponding unitary transformation can be given as,

$$U_{QFT} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} e^{\frac{2\pi i jk}{\sqrt{N}}} |k\rangle \langle j|$$

Expanding the summations and decomposing k into fractional binary notation and simplifying the equation we get(7),

$$QFT\,|j\rangle = \frac{1}{\sqrt{2^n}} \left(|0\rangle + ^{2\pi i j 2^{-1}} |1\rangle \right) \left(|0\rangle + ^{2\pi i j 2^{-2}} |1\rangle \right) \ldots \left(|0\rangle + ^{2\pi i j 2^{-n}} |1\rangle \right)$$

The above unitary transformation can be broken down into tensor product of series of Hadamard and Controlled Phase quantum gates, which is described in the algorithm flowchart(7).

3. Signal Processing with QFT

This section compares the results obtained by digital signal processing of certain audio files using classical techniques like Fast Fourier Transform from numpy package and QFT implemented on a quantum circuit with Qiskit software package(5).

QFT Algorithm :

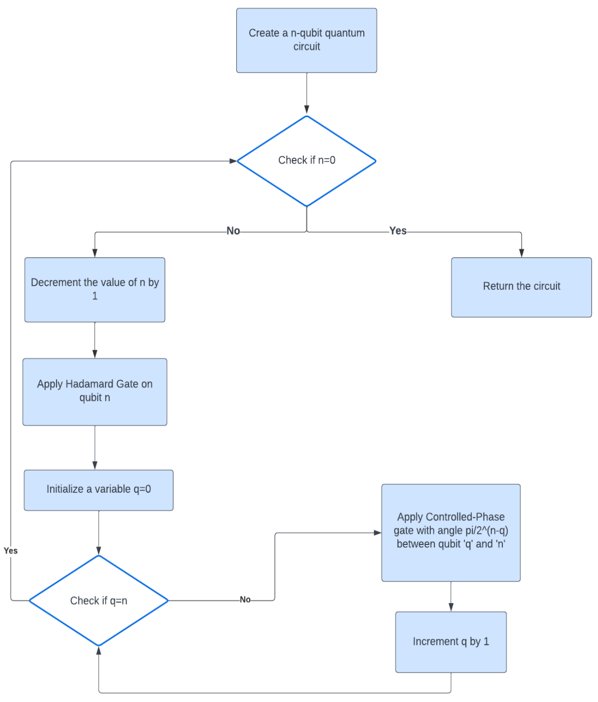
Create a n-qubit quantum circuit
Check if n=0
No
Yes
Decrement the value of n by 1
Return the circuit
Apply Hadamard Gate on qubit n
Initialize a variable q=0
Apply Controlled-Phase gate with angle pi/2^(n-q) between qubit 'q' and 'n'
Yes
Check if q=n
No
Increment q by 1

Figure 2: Flowchart for QFT Algorithm

3.1. Obtaining the Frequency Spectrum

Frequency spectrum for an audio sample can be obtained by taking the modulus of the fourier coefficients calculated by digital signal processing, which is exactly the case for FFT. While in the case of QFT, I chose to run the quantum circuit in a *ibmq qasm simulator* for 10000 shots. This would give me a list of probabilities, which are the modulus square of the fourier coefficients. In the figure 2-4, are the results of the frequency spectrum obtained from using QFT with three different audio data, in comparison to the frequency spectrum obtained from classical FFT algorithm.

Figure 2 is obtained for an audio signal of a musical instrument *cello*, with its data encoded(4) into a 12 qubit quantum circuit. Whereas figure 3 and 4 are obtained for an audio signal, with its data encoded into a 15 qubit quantum circuit. The observation to be noted here is that, the frequency spectrum obtained from QFT is equivalent to that of FFT's, but not exactly equal. This behaviour is expected, as plots shown are the probabilities extracted from the measurement outcomes of the circuit executed with shots = 10000, which ideally should be infinity. Another point to note is that the frequency corresponding to the maximum intensity remains same for both FFT and QFT, thus making QFT reliable for obtaining frequency spectrum(2). Infact, the top five frequencies of maximum intensities extracted from both the spectrum were the same.
The total power of the audio signal can be obtained from the

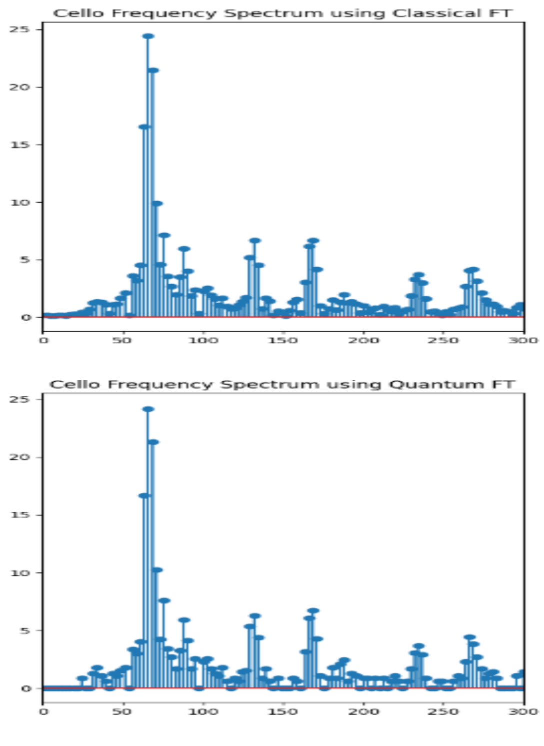

Figure 3: Fourier Spectrum of an audio from Cello musical instrument with a size of 2^{12} sample points

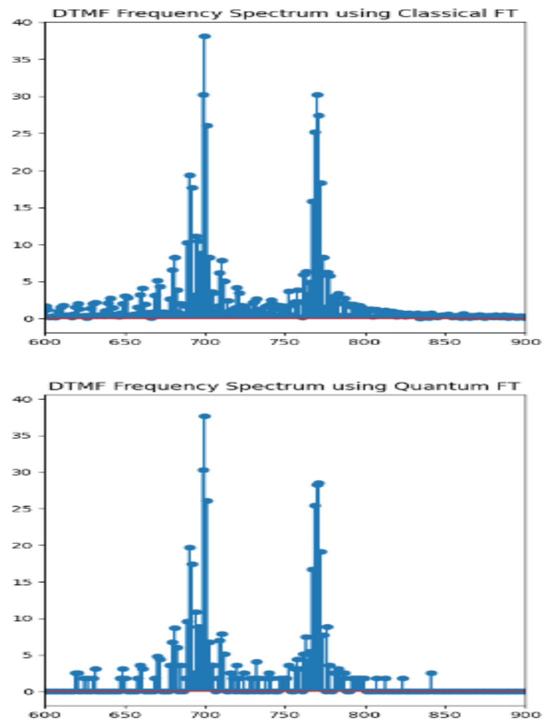

Figure 5: Fourier Spectrum of DTMF audio signal with a size of 2^{15} sample points

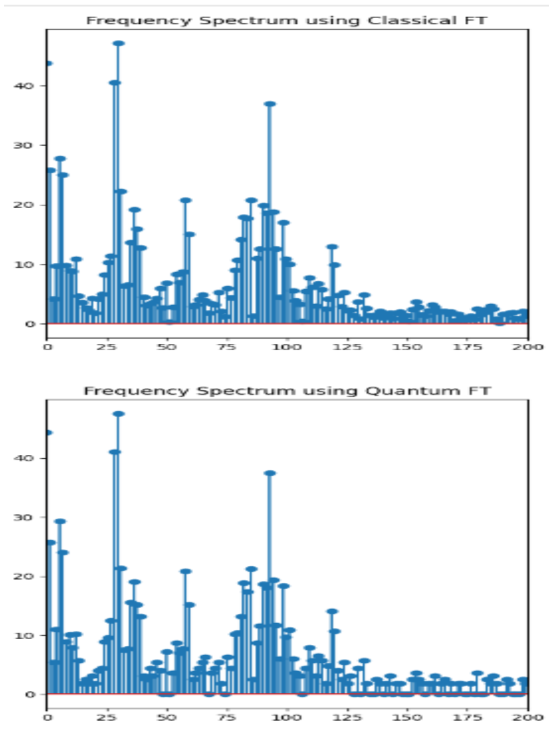

Figure 4: Fourier Spectrum of an audio signal with a size of 2^{15} sample points

frequency spectrum by,

$$P_T = \sum_{k}^{N-1} |y_k|^2$$

where y_k are the fourier coefficients. Considering the point that, the audio data is normalized, the total power of the audio signal calculated from the FFT's and QFT's frequency spectrum should be equal to length of the data, which has been verified to be exactly correct.

3.2. *Obtaining the Spectrogram*

Spectrogram is another important technique which can be implemented with the use of fourier transform algorithm. The spectrogram is generated by selecting a window of a specific duration and computing the Fourier spectrum for the data points within this window. The window is then shifted by a constant fraction of its length, as determined by the slide parameter. Subsequently, the Fourier spectrum is recalculated. This iterative process continues until the window has traversed the entire length of the data. As a result, the spectrogram illustrates the frequencies present in the data over time, capturing changes as the window is shifted. In the figure 5 and 6, the spectrogram obtained from using QFT and FFT algorithm are compared for two different audio files.

The spectrograms from QFT shown in the figure 5 and 6, are obtained by simulating the output statevector of the quantum circuit. This can be done by running the quantum circuit in a IBM's *statevector simulator* or by calling the *Statevector instance* from quantum info module in Qiskit. The observation to be noted here is that, the QFT algorithm provides the accurate output as that of classical FFT algorithm. Let me make a special remark for figure 6, which is obtained for Dual-Tone Multi-Frequency (DTMF)(3) audio file. DTMF is a signaling system used in telecommunication and telephone systems to represent

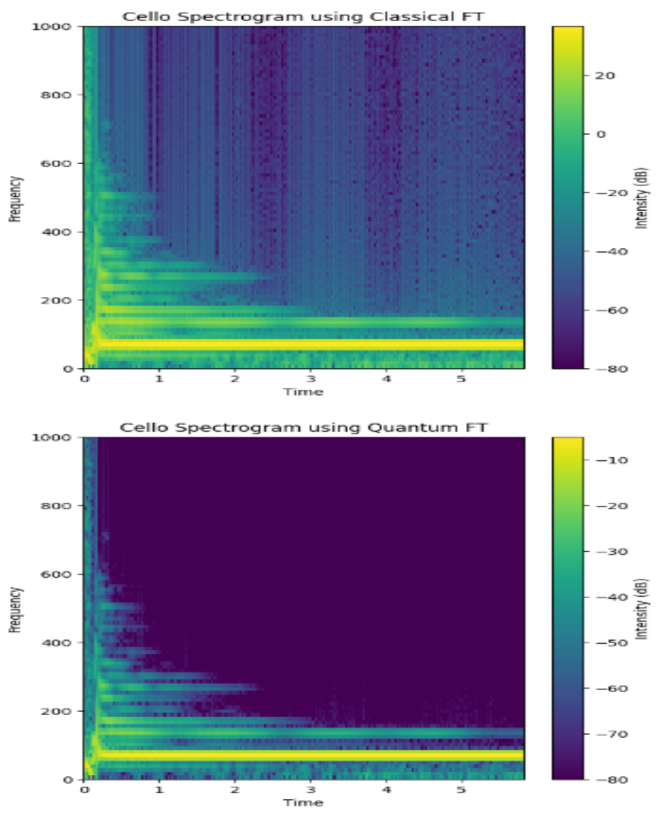

Figure 6: Spectrogram of Cello audio with a size of 2^{12} sample points

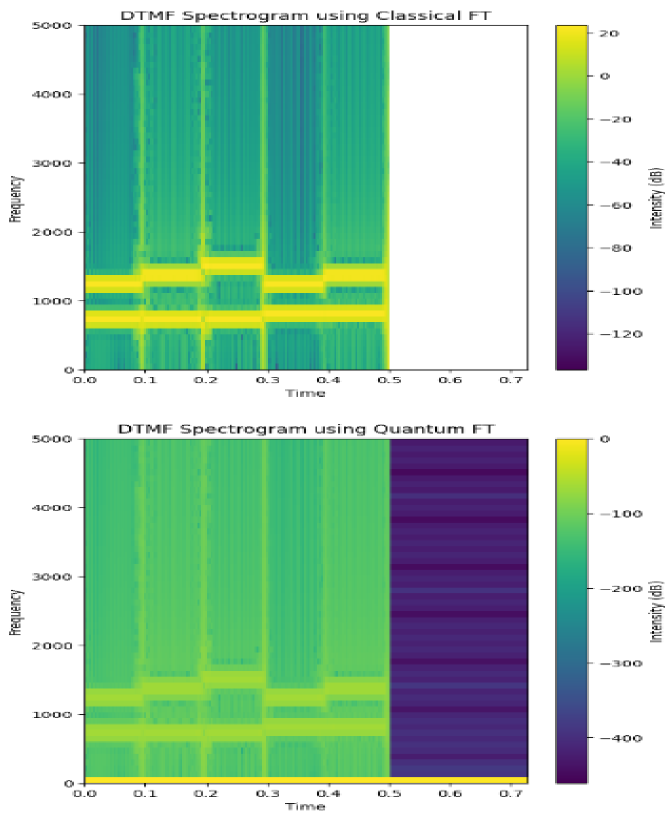

Figure 7: Spectrogram of DTMF audio signal with a size of 2^9 sample points

digits, letters, and symbols. It assigns a unique pair of audio frequencies (one high and one low) to each key on a telephone keypad. When a key is pressed, the corresponding pair of frequencies is transmitted, and the receiver can decode these frequencies to identify the pressed key. The DTMF audio file used here was obtained for telephone keys 1-2-3-4-5 pressed in the same order. Both the spectrograms, classical and quantum, for this file, make the clear distinction of the frequencies, by showcasing exactly two types of frequencies at a given time. Thus, here QFT is shown to be as reliable as FFT to decode the DTMF audio.
An important point to note here is that, while obtaining the above spectrograms from QFT algorithm, I have not applied measurements or ran the quantum circuit in a actual quantum computer, because the statevector (in our case this is the array of complex fourier coefficients) information is lost when the measurement is done, as it only provides us with information on modulus square of the fourier coefficients like in subsection before. Thus making it not possible to obtain the phase information which is very important to build a spectrogram. There is a work-around for this consequence, which is the process of *Quantum State Tomography* (QST). This is a process which reconstructs the density matrix or the statevector that fully characterizes the quantum circuit by measuring it in different sets of basis. There are ongoing advancements in QST algorithms to enhance the accuracy and efficiency of state reconstruction. This subsection can be viewed as a proof of concept or benchmarking for the QFT algorithm.

3.3. Filtering and Reconstructing the audio signal

Filtering is another signal processing technique, that helps us get the desired information from signal by eliminating the unwanted noisy part. Filtering, for example applying *low pass filter* on a audio can be realised by first obtaining the frequency spectrum of the signal and then eliminating the frequencies above desired cut-off frequency from it. Following it, one can use inverse fourier transform to obtain the filtered signal back. Figure 8 realises the filtered frequency spectrum of the Cello audio signal for the cutoff frequency of 200Hz. One can observe that spectrum from QFT is almost exactly similar to that of FFT. Figure 9 realises the audio reconstructed from the filtered frequency spectrum obtained by classical FFT algorithm. I have used the inverse FFT to obtain it. The absence of higher frequency vibrations, is clearly profound in comparison of actual and filtered signal plots.

An important point to note is that, filtering is not an unitary operation, because of which implementing it in the quantum circuit is not possible. In such situations, the hybrid quantum-classical computation paves a way for faster and efficient results.

4. Conclusion

In this paper, I have reviewed the quantum Fourier transform in real world application of digital signal processing. All the

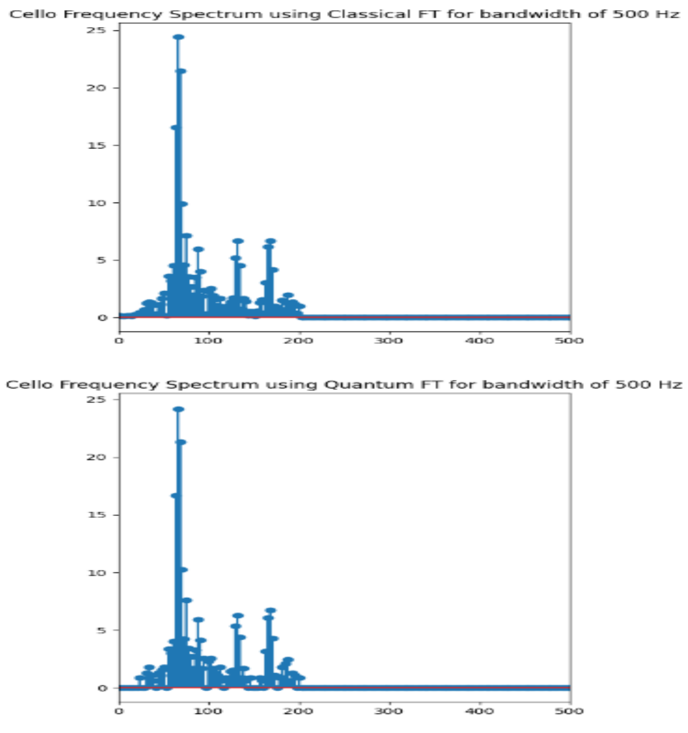

Figure 8: Filtered Cello frequency spectrum

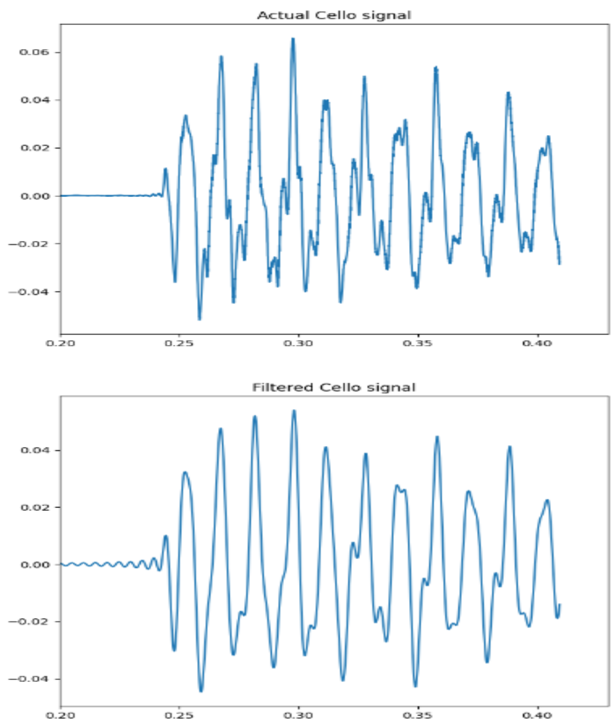

Figure 9: Reconstructed Cello audio signal

above computational work has been done in Python programming language and with the help of basic packages like Numpy, Qiskit and quantumaudio which is developed and maintained by the quantum computer music team at the Interdisciplinary Centre for Computer Music Research (ICCMR), University of Plymouth, UK. The computational work can be found in my github repo : *https://github.com/Ak-ash22/qft_symphony*.

The QFT algorithm helps us to realise the discrete Fourier transform as a tensor products of unitary quantum gates like Hadamard and Controlled-Phase gate, due to which it is crucial to many algorithms in quantum processing. The classical FFT algorithm, is one of the efficient algorithm for computing Discrete Fourier Transform and also its inverse. The FFT algorithms provides a speedup to $O(N\log(N))$ from the naive $O(N^2)$ complexity of DFT. The gate complexity of QFT used here is of $O(N^2)$, but there are very efficient approximate versions which need only $O(N\log(N))$ gates possibly giving more speedup (6). One should note that this excludes the complexity of initializing the data onto a quantum circuit. This paper provides the results for using QFT in digital signal processing, which are obtained by running the circuits on a ideal noise free quantum simulator. It is true that, since we are currently in the NISQ (*Noisy Intermediate Scale-Quantum*) era, the results would be noisy if run on actual quantum computers and thus might require error correction techniques. Despite the challenges in the NISQ era, there have been rapid advancements in quantum hardware to realize the full potential of quantum computing. The exploration of quantum algorithms such as QFT in real world applications, serves as a crucial step in understanding the capabilities and limitations of quantum technologies.

References

[1] Qc — quantum fourier transform. quantum fourier transform is a very… — by jonathan hui — medium. `https://jonathan-hui.medium.com/qc-quantum-fourier-transform-45436f90a43`. (Accessed on 12/22/2023).

[2] Quantum computing to find frequencies in an audio file. `https://sarangzambare.github.io/jekyll/update/2020/06/13/quantum-frequencies.html`. (Accessed on 12/22/2023).

[3] Wikipedia-dual-tone.pdf. `https://www.ece.iastate.edu/~alexs/classes/2021_Spring_575/code/19_Audio/02_Phone_Digits/Wikipedia-Dual-Tone.pdf`. (Accessed on 12/22/2023).

[4] Paulo Vitor Itaboraí and Eduardo Reck Miranda. Quantum representations of sound: from mechanical waves to quantum circuits. In *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, pages 223–274. Springer, 2022.

[5] Shlomo Kashani, Maryam Alqasemi, and Jacob Hammond. A quantum fourier transform (qft) based note detection algorithm. *arXiv preprint arXiv:2204.11775*, 2022.

[6] Yunseong Nam, Yuan Su, and Dmitri Maslov. Approximate quantum fourier transform with o (n log (n)) t gates. *NPJ Quantum Information*, 6(1):26, 2020.

[7] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.